Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

On normalisation

F. van Raamsdonk and P. Severi

Computer Science/Department of Software Technology

CS-R9545 1995

Report CS-R9545
ISSN 0169-118X

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

On Normalisation

Femke van Raamsdonk
CWI
P.O. Box 94079, 1090 GB Amsterdam
The Netherlands
femke@cwi.nl

Paula Severi
Department of Mathematics and Computing Science
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven
The Netherlands
severi@win.tue.nl

Abstract

Using a characterisation of strongly normalising λ-terms, we give new and simple proofs of the following:

1. all developments and superdevelopments are finite,
2. a certain rewrite strategy is perpetual,
3. a certain rewrite strategy is maximal and thus perpetual,
4. simply typed λ-calculus is strongly normalising.

AMS Subject Classification (1991): 03B40, 03D70.
CR Subject Classification (1991): F.3.3, F.4.1.
Keywords & Phrases: λ-calculus, normalisation, perpetual strategies.
Note: The research of the first author is supported by NWO/SION project 612-316-606. This report is also available as Computing Science Report 95/20, Eindhoven University of Technology.

1. Introduction

This paper represents an effort to shed some more light on various results concerning normalisation in λ-calculus. We deal with λ-calculus with only β-reduction.

As a first step towards a better understanding we characterise both the set of weakly normalising terms and the set of strongly normalising terms. Remember that a term M is said to be *weakly normalising* if there is a rewrite sequence starting in M that eventually ends in a normal form, and that a term M is said to be *strongly normalising* if all rewrite sequences starting in M end eventually in a normal form.

To give a characterisation of all weakly normalising terms is actually rather easy: a weakly normalising term is a normal form or can be obtained as the result of some expansion starting in a normal form. Or, to put it slightly differently, the set of all weakly normalising terms is exactly the smallest set of all normal forms closed under expansion.

A specialisation of this idea yields a quite elegant characterisation of the strongly normalising terms, in the form of an inductively defined set denoted as $\mathcal{SN}$. The definition of $\mathcal{SN}$ can be found in section 3. This set can be viewed in different ways. From the point of view of rewriting, it is the closure under expansion of the set of normal forms, where expansion is subject to two restrictions. These restrictions are the following: first, the argument of the redex introduced by the expansion step should be in the set of strongly normalising terms, and second, the expansion step should yield a new head redex or a new outermost redex in a term without a head redex. Further, those familiar with saturated sets will certainly recognise one of the clauses of the definition of $\mathcal{SN}$ as a defining property of a saturated set.

The interesting thing of the definition of the set $\mathcal{SN}$ is that it permits to give new proofs of important results concerning normalisation in λ-calculus, like the Finite Developments Theorem and the fact that all simply typable terms are normalising. In most cases the new proofs are essentially simpler than already existing ones. Moreover, we feel that it is important to have different proofs of important results because they may help us to understand not only *the mechanics* of the proofs of the results but also *the reasons* for their validity.

The remainder of the paper is organised as follows.

In Section 3, we prove that the set $\mathcal{SN}$ characterises the set of β-strongly normalising terms. We also give another characterisation of the set of all strongly normalising terms.

In Section 4 we give a short and simple proof of finiteness of developments. It is different from existing proofs, like for instance the one using a decreasing labelling or the elegant proof given by De Vrijer. Our proof makes use of expansion.

In Section 5 we prove that all superdevelopments are finite. Superdevelopments are developments in which redexes that are created 'upwards' during reduction may be contracted.

In Section 6 we give a new proof of the fact that the strategy F_{bk} defined by Bergstra and Klop, is perpetual (meaning that it yields an infinite rewrite sequence whenever possible). Further we prove that the strategy F_∞ defined by Barendregt, Bergstra, Klop and Volken is perpetual.

For the strategy F_∞, we prove in Section 7 that it is not only perpetual but also maximal. That is, it yields the longest possible reduction to normal form whenever the initial term is strongly normalising, and an infinite rewrite sequence if possible. This is done by computing the length of the rewrite sequence to the normal form.

In Section 8 we prove that simply typed λ-calculus is strongly normalising using our characterisation of the strongly normalising terms. The definition of $\mathcal{SN}$ clearly recalls the definition of saturation, see [Tai67] and [Gir72].

In Section 9 we consider λ-calculus with intersection types. The set of strongly normalising terms is the set of terms that are typable in $\lambda\wedge$. We felt obliged to compare both characterisations and give a direct proof of the fact that the set $\mathcal{SN}$ coincides with the set of typable terms in $\lambda\wedge$.

Finally, we discuss related work in section 10.

We start by reviewing some notation and by formalising the concept of lifting of rewrite sequences for the case of Abstract Rewriting Systems.

2. Preliminaries

Notation. We assume familiarity with λ-calculus and just fix some (mostly standard) notation.

The set of λ-terms is denoted by Λ. We write $x, y, z, \ldots$ for variables and $M, N, P, Q, \ldots$ for terms. We assume α-conversion to be applied whenever necessary. The symbol $[\,]$ is used to denote a hole in a term. A term with one or more occurrences of $[\,]$ is called a *context* and is denoted by $C[\,]$. The term obtained by replacing in a context $C[\,]$ the occurrences of $[\,]$ by a term M is denoted by $C[M]$. If not specified otherwise, in this paper a context $C[\,]$ is supposed to contain one occurrence of $[\,]$. We suppose a term to contain no occurrences of $[\,]$.

The set of free variables of a term M is denoted by $\mathsf{FV}(M)$ and its set of bound variables is denoted by $\mathsf{BV}(M)$.

We consider λ-calculus with β-reduction generated by the β-reduction rule, that is given as $(\lambda x.M)N \to M[x := N]$. We denote the β-reduction relation by $\to_\beta$ or by $\stackrel{\phi}{\to}_\beta$ if we want to specify that the β-rewrite step is obtained by contracting a β-redex at position ϕ. The reflexive-transitive closure of $\to_\beta$ ($\stackrel{\phi}{\to}_\beta$) is denoted by $\twoheadrightarrow_\beta$ ($\stackrel{\phi}{\twoheadrightarrow}_\beta$). Syntactic equality is denoted by $=$.

The set of normal forms is denoted by $\mathcal{NF}$ and $\mathsf{nf}(M)$ denotes the normal form of a term M.

A term M is said to be *strongly normalising* if every rewrite sequence ends after finitely many steps in a normal form. A term M is said to be *weakly normalising* if there is a rewrite sequence starting at M that ends in a normal form.

Lifting. In rewriting one often makes use of terms that are decorated, for instance by labels. Also the rewrite relation can be decorated, in the sense that the decoration is a part of the pattern of the rewrite rule. Erasure of decoration is used in order to switch from the rewriting system with decorations to the original rewriting system without decorations. The thus obtained correspondence consists in fact of two parts: the correspondence between terms and decorated terms and the correspondence between steps and decorated steps. A decorated rewrite sequence is often called a *lifting* of the rewrite sequence in the original rewriting systems, that is obtained by erasing all decorations. In this paper we shall encounter two different ways of lifting a β-rewrite sequence in λ-calculus. Here we formalise the concept of lifting for the general case, that is, for Abstract Rewriting Systems that are enriched with some more structure.

DEFINITION 2.1. An *Abstract Rewriting System* is a pair $(A, \rightarrow)$ consisting of a set A and a relation $\rightarrow \subseteq A \times A$.

NOTATION 2.2. Abstract Rewriting Systems are denoted by $\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots$.

DEFINITION 2.3. A rewrite step in an Abstract Rewriting System $\mathcal{A} = (A, \rightarrow)$ is a pair (a, b) of elements of A with $(a, b) \in \rightarrow$.

NOTATION 2.4. We write $a \rightarrow b$ instead of $(a, b) \in \rightarrow$.

In rewriting systems where the rewrite relation is induced by a set of rewrite rules, it may happen that different rewrite steps between terms a and b exist. This is for instance the case in λ-calculus: there are two ways to rewrite $(\lambda x.x)((\lambda x.x)y)$ to $(\lambda x.x)y$. This cannot be expressed in Abstract Rewriting Systems. For the concept of correspondence between rewrite sequences we have in mind it is important that not only a correspondence between terms but also between rewrite steps can be expressed. Therefore we will consider Abstract Rewriting Systems enriched with some more structure, called *indexed Abstract Rewriting Systems*. The idea is to view $\rightarrow$ as a collection of partial functions on A, written as $\{\rightarrow_i\}_{i \in I}$. For instance λ-calculus can be seen as the set of λ-terms Λ and a collection of partial functions $\{\stackrel{\phi}{\rightarrow}_\beta\}_\phi$. It is clear why the functions are partial: for instance a step $\stackrel{\epsilon}{\rightarrow}_\beta$ is not defined on every λ-term. We proceed by giving the definition of an indexed Abstract Rewriting System.

DEFINITION 2.5. An *indexed Abstract Rewriting System* is a triple $(A, I, \{\rightarrow_i\}_{i \in I})$ consisting of a set A, a set of indices I and an indexed set of partial functions $\{\rightarrow_i\}_{i \in I}$ from A to A.

The definition of a rewrite step in an indexed Abstract Rewriting System differs a bit from the one in an Abstract Rewriting System.

DEFINITION 2.6. A rewrite step in an indexed Abstract Rewriting System $\mathcal{A} = (A, I, \{\rightarrow_i\}_{i \in I})$ is a triple consisting of two elements a, b of A and an element $\rightarrow_i$ of $\{\rightarrow_i\}_{i \in I}$ such that $(a, b) \in \rightarrow_i$.

NOTATION 2.7. We write $a \rightarrow_i b$ for $(a, b) \in \rightarrow_i$.

Note that in an indexed Abstract Rewriting System $a \rightarrow_i b$ and $a \rightarrow_i c$ implies $b = c$. It is possible to have $a \rightarrow_i b$ and $a \rightarrow_j b$ with $i \neq j$.

NOTATION 2.8. We use the notation $\rightarrow_i (a) = b$ to denote that $\rightarrow_i$ is defined on a and that the result of applying $\rightarrow_i$ to a equals b.

The index i of a rewrite step $a \to_i b$ can be considered to be the name of the rewrite step. In the case of term rewriting, taking redexes (that is, pairs consisting of a position and a rewrite rule) as indexes yields an instance of an indexed Abstract Rewriting System.

We now give a formal definition of a rewrite sequence in an indexed Abstract Rewriting System. It can be generalised to the case of Abstract Rewriting Systems, however, we don't need a so general definition in the present paper. See [Oos94] for a general definition of conversion.

DEFINITION 2.9. Let $\mathcal{A} = (A, I, \{\to_i\}_{i\in I})$ be an indexed Abstract Rewriting System. A *rewrite sequence* of length α starting in a is a triple (a, α, σ) satisfying the following:

1. $a \in A$,
2. $\alpha \leq \omega$,
3. $\sigma : \alpha \to I$ is a mapping that defines a sequence $\{a_n\}_{n\in\alpha}$ as follows:
   (a) $a_0 = a$,
   (b) $a_n = \to_{\sigma(n)} (a_{n-1})$ for all $n \in \alpha\backslash\{0\}$

NOTATION 2.10. We often denote a rewrite sequence σ as in the previous definition by $\sigma : a_0 \to_{\sigma(1)} a_1 \to_{\sigma(2)} \ldots$.

We now define the concept of morphism between Abstract Rewriting Systems. It will be used to formalise the notion of correspondence between two rewrite sequences.

DEFINITION 2.11. A *morphism* between indexed Abstract Rewriting Systems $\mathcal{A} = (A, I, \{\to_i\}_{i\in I})$ and $\mathcal{B} = (B, J, \{\to_j\}_{j\in J})$ is a pair of mappings $f = (f_0, f_1)$ with

$$\begin{aligned} f_0 :&\quad A \to B \\ f_1 :&\quad I \to J \end{aligned}$$

such that $f_0(\to_i (a)) = \to_{f_1(i)} (f_0(a))$ for all $a \in A$ such that $\to_i (a)$ is defined.

Note that in the equality $f_0(\to_i (a)) = \to_{f_1(i)} (f_0(a))$ in the previous definition it may occur that $\to_{f_1(i)} (f_0(a))$ is defined but $f_0(\to_i (a))$ isn't.

We often write f for both f_0 and f_1.

Let $\sigma : a_0 \to_{m_1} a_1 \to_{m_2} a_2 \to_{m_3} \ldots$ be a rewrite sequence in an indexed Abstract Rewriting System $\mathcal{A} = (A, I, \{\to_i\}_{i\in I})$. Let $f : \mathcal{A} \to \mathcal{B}$ be a morphism between indexed Abstract Rewriting Systems. We denote by $f(\sigma)$ the rewrite sequence $f(\sigma) : f_0(a_0) \to_{f_1(m_1)} f(a_1) \to_{f_1(m_2)} f_0(a_2) \to_{f_1(m_3)} \ldots$. Note that by the definition of a morphism this indeed is a well-defined rewrite sequence in $\mathcal{B}$.

DEFINITION 2.12. Let $f : \mathcal{A} \to \mathcal{B}$ be a morphism. A rewrite sequence σ in an indexed Abstract Rewriting System $\mathcal{A} = (A, I, \{\to_i\}_{i\in I})$ is an *f-lifting* of a rewrite sequence ρ in an indexed Abstract Rewriting System $\mathcal{B} = (B, J, \{\to_j\}_{j\in J})$ if $f(\sigma) = \rho$.

Examples of liftings can be found in Section 4 and in Section 5.

3. A Characterisation of Strongly Normalising λ-terms

In this section we characterise the set of λ-terms that are strongly normalising.

The characterisation.

Definition 3.1. The set $\mathcal{SN}$ is the smallest set of λ-terms satisfying the following:

1. if x is a variable and $M_1, \ldots, M_n \in \mathcal{SN}$ for some $n \geq 0$, then $xM_1 \ldots M_n \in \mathcal{SN}$,
2. if $M \in \mathcal{SN}$ then $\lambda x.M \in \mathcal{SN}$,
3. if $M[x := N]P_1 \ldots P_n \in \mathcal{SN}$ and $N \in \mathcal{SN}$, then $(\lambda x.M)NP_1 \ldots P_n \in \mathcal{SN}$.

We prove that the set $\mathcal{SN}$ characterises the strongly normalising terms.

Theorem 3.2. *M is strongly normalising if and only if $M \in \mathcal{SN}$.*

Proof.

$\Rightarrow$. Let M be a strongly normalising term. The proof proceeds by induction on the pair $(\mathsf{maxred}(M), M)$, lexicographically ordered by the usual ordering on $\mathbb{N}$ and the subterm ordering. Here we denote by $\mathsf{maxred}(M)$ the maximum length of a reduction from M to normal form.

The base case is trivial since it is easy to see that all normal forms are in $\mathcal{SN}$.

Suppose the maximal reduction of M to normal form takes $k+1$ steps. Let $M = \lambda x_1 \ldots \lambda x_n.PQ_1 \ldots Q_m$. There are two cases.

Case 1. $P = y$. Then the normal form of M is of the form $\lambda x_1 \ldots \lambda x_n.yQ'_1 \ldots Q'_m$ with $Q_i \twoheadrightarrow Q'_i$ for $i = 1, \ldots, m$. By induction hypothesis, $Q_1 \in \mathcal{SN}, \ldots, Q_m \in \mathcal{SN}$. By the first and second clause of the definition of $\mathcal{SN}$, we have $M = \lambda x_1 \ldots \lambda x_n.yQ_1 \ldots Q_m \in \mathcal{SN}$.

Case 2. $P = \lambda y.P_0$. We have $M = \lambda x_1 \ldots \lambda x_n.(\lambda y.P_0)Q_1Q_2 \ldots Q_m \to \lambda x_1 \ldots \lambda x_n.P_0[y := Q_1]Q_2 \ldots Q_m$. By induction hypothesis, $\lambda x_1 \ldots \lambda x_n.P_0[y := Q_1]Q_2 \ldots Q_m \in \mathcal{SN}$. Also by induction hypothesis, $Q_1 \in \mathcal{SN}$. By the last clause of the definition of $\mathcal{SN}$, we have $M = \lambda x_1 \ldots \lambda x_n.(\lambda y.P_0)Q_1 \ldots Q_m \in \mathcal{SN}$.

$\Leftarrow$. Suppose $M \in \mathcal{SN}$. We prove by induction on the derivation of $M \in \mathcal{SN}$ that M is strongly normalising.

1. If $M = xM_1 \ldots M_n$ with $M_1, \ldots, M_n \in \mathcal{SN}$, then the statement follows easily by induction hypothesis.
2. If $M = \lambda x.M_0$ with $M_0 \in \mathcal{SN}$, then by induction hypothesis M_0 is strongly normalising. Then also $M = \lambda x.M_0$ is strongly normalising.
3. Let $M = (\lambda x.M_0)M_1M_2 \ldots M_n$ with $M_0[x := M_1]M_2 \ldots M_n \in \mathcal{SN}$ and $M_1 \in \mathcal{SN}$. Consider an arbitrary rewrite sequence $\rho : M = P_0 \to_\beta P_1 \to_\beta P_2 \to_\beta \ldots$ starting in M. There are two possibilities: in ρ either the head redex of M is contracted or the head redex of M is not contracted.

In the first case, there is an i such that $P_i = M'_0[x := M'_1]M'_2 \dots M'_n$, with $M_0 \twoheadrightarrow M'_0, \dots, M_n \twoheadrightarrow M'_n$. Then P_i is a result of rewriting the term $M_0[x := M_1]M_2 \dots M_n$. The latter is by induction hypothesis strongly normalising. Hence P_i is strongly normalising so ρ is finite.

In the second case, all terms in ρ are of the form $(\lambda x.M'_0)M'_1M'_2 \dots M'_n$ with $M_0 \twoheadrightarrow M'_0, \dots, M_n \twoheadrightarrow M'_n$. By induction hypothesis, the term $M_0[x := M_1]M_2 \dots M_n$ is strongly normalising. Therefore $M_0, M_2, \dots, M_n$ are strongly normalising. Moreover, we have by induction hypothesis that M_1 is strongly normalising. Hence all the terms in the rewrite sequence are strongly normalising and hence ρ is finite.

□

Background. We would like to point out the considerations motivating the previous definition.

An easy observation is that the set that contains all normal forms and that is closed under expansion is exactly the set of all weakly normalising terms. So we have the following definition.

Definition 3.3. The set $\mathcal{W}$ is the smallest set of λ-terms satisfying the following:

1. all normal forms are in $\mathcal{W}$,
2. if $C[P[x := Q]] \in \mathcal{W}$, then $C[(\lambda x.P)Q] \in \mathcal{W}$.

The first naive attempt to obtain the set of all strongly normalising terms, is to add the requirement that the argument of the redex introduced by the expansion is strongly normalising. The thus defined set $\mathcal{S}$ is the smallest set that satisfies

1. all normal forms are in $\mathcal{S}$,
2. if $C[P[x := Q]] \in \mathcal{S}$ and $Q \in \mathcal{S}$, them $C[(\lambda x.P)Q] \in \mathcal{S}$.

However, it is easy to see that the weakly but not strongly normalising term $(\lambda x.(\lambda y.z)(xx))(\lambda y.(yy))$ belongs to $\mathcal{S}$. The problem is that expansions cannot be allowed to take place just everywhere. If the expansion as in the second clause of the definition of $\mathcal{S}$ above is required to create a *head redex* or, if the result of the expansion doesn't contain a head redex, to create an *outermost redex*, then we indeed obtain the set of all strongly normalising terms.

Definition 3.4. The set $\mathcal{O}$ of contexts with a hole at a head or outermost position is defined as the minimal set that satisfies

1. if $C[\,] \in \mathcal{O}$ then $x\ M_1\ \dots C[\,] \dots M_n \in \mathcal{O}$,
2. if $C[\,] \in \mathcal{O}$ then $\lambda x.C[\,] \in \mathcal{O}$,
3. $[\,]P_1 \dots P_n \in \mathcal{O}$.

Definition 3.5. The set $\mathcal{SN}'$ is defined as the smallest set that satisfies

1. all normal forms are in $\mathcal{SN}'$,
2. if $C[P[x := Q]] \in \mathcal{SN}'$, $Q \in \mathcal{SN}'$ and $C[\,] \in \mathcal{O}$, then $C[(\lambda x.P)Q] \in \mathcal{SN}'$.

It is not difficult to show that $\mathcal{SN}' = \mathcal{SN}$.

Through the paper we use the first characterisation of the strongly normalising terms because besides being easier to handle it seems more natural, as it clearly recalls the notion of saturated sets.

Inductive Definitions. We often describe a set by induction as the smallest set closed under some set of rules (see [Acz77] and [Ter94]). This is the way we define, for example, the class of theorems of a given system. And this is also the way we have defined the set $\mathcal{SN}$ in Definition 3.1.

We could have described the set $\mathcal{SN}$ by induction giving a monotone operator instead of giving a set of rules. In this case, we should have defined the set $\mathcal{SN}$ as the least fixed point of the operator $H : \mathcal{P}(\Lambda) \to \mathcal{P}(\Lambda)$ given by:

$$\begin{array}{rcl} H(X) & = & \{xM_1 \ldots M_n) \mid M_i \in X \text{ for all } i = 1, \ldots, n \text{ and } x \text{ a variable } \} \cup \\ & = & \{(\lambda x.M) \;\mid\; M \in X\} \cup \\ & = & \{(\lambda x.M)NP_1 \ldots P_n \;\mid\; M[x := N]P_1 \ldots P_n \in X \;\text{ and }\; N \in X\} \end{array}$$

Another possibility is to define the set $\mathcal{SN}$ as the well-founded part of the order $\prec \subset \Lambda \times \Lambda$ defined by:

1. $M_i \prec xM_1 \ldots M_n$ for all $i = 1, \ldots, n$,
2. $M \prec \lambda x.M$,
3. $M[x := N]P_1 \ldots P_n \prec (\lambda x.M)NP_1 \ldots P_n$,
4. $N \prec (\lambda x.M)NP_1 \ldots P_n$.

In [Acz77], these three ways of giving a definition by induction, that is giving a set of deterministic rules or clauses, giving a monotone operator and giving an order are proved to be equivalent.

4. Finite Developments

A *development* is a rewrite sequence in which only descendants of redexes that are present in the initial term may be contracted. In this section we give a new and short proof of the fact that in λ-calculus all β-developments terminate.

Usually, β-developments are defined via a set of underlined λ-terms and an underlined β-reduction rule. We shortly recall these definitions, for a complete formal treatment see [Bar84].

Definition 4.1. The set of underlined λ-terms $\underline{\Lambda}$ is the smallest set satisfying the following:

1. $x \in \underline{\Lambda}$ for every variable x,
2. if $M \in \underline{\Lambda}$, then $\lambda x.M \in \underline{\Lambda}$,

3. if $M \in \underline{\Lambda}$ and $N \in \underline{\Lambda}$, then $MN \in \underline{\Lambda}$,

4. if $M \in \underline{\Lambda}$ and $N \in \underline{\Lambda}$, then $(\underline{\lambda} x.M)N \in \underline{\Lambda}$.

The $\underline{\beta}$-rewrite relation is defined as follows:

$$(\underline{\lambda} x.M)N \to M[x := N]$$

Note that $\underline{\Lambda}$ is closed under $\underline{\beta}$-rewriting.
In order to be able to switch between Λ and $\underline{\Lambda}$ a mapping $\mathcal{E}$ that erases underlinings is defined.

DEFINITION 4.2.

1. The mapping $\mathcal{E}_0 : \underline{\Lambda} \to \Lambda$ is defined by induction on the definition of $\underline{\Lambda}$.
   (a) $\mathcal{E}_0(x) = x$,
   (b) $\mathcal{E}_0(\lambda x.M) = \lambda x.\mathcal{E}_0(M)$,
   (c) $\mathcal{E}_0(MN) = \mathcal{E}_0(M)\mathcal{E}_0(N)$,
   (d) $\mathcal{E}_0((\underline{\lambda} x.M)N) = (\lambda x.\mathcal{E}_0(M))\mathcal{E}_0(N)$

2. The mapping $\mathcal{E}_1 : \{\underline{\beta}\} \times \{0,1\}^* \to \{\beta\} \times \{0,1\}^*$ is defined by $\mathcal{E}_1(\underline{\beta}, \phi) = (\beta, \phi)$.

Since it is clear that $M \xrightarrow{\phi}_{\underline{\beta}} N$ in $\underline{\Lambda}$ implies $\mathcal{E}(M) \xrightarrow{\phi}_{\beta} \mathcal{E}(N)$ in Λ, the proof of the following lemma is straightforward.

LEMMA 4.3. The mapping $\mathcal{E} = (\mathcal{E}_0, \mathcal{E}_1)$ is a morphism from $(\underline{\Lambda},\ \{\underline{\beta}\} \times \{0,1\}^*,\ \{\xrightarrow{\phi}_{\underline{\beta}}\}_\phi)$ to $(\Lambda,\ \{\beta\} \times \{0,1\}^*,\ \{\xrightarrow{\phi}_{\beta}\}_\phi)$.

The definition of a development is as follows.

DEFINITION 4.4. A rewrite sequence $\sigma : M \twoheadrightarrow_\beta N$ in Λ is a *development* if there is a rewrite sequence ρ in $\underline{\Lambda}$ that is an $\mathcal{E}$-lifting of σ.

We give a new and short proof of finiteness of developments by considering another inductive definition of the set of all underlined λ-terms. Like the set $\mathcal{SN}$ of strongly normalising λ-terms, this definition makes essential use of expansion.

DEFINITION 4.5. The set $\mathcal{D}$ is the smallest set of λ-terms satisfying

1. $x \in \mathcal{D}$ for all variables x,

2. if $M \in \mathcal{D}$, then $\lambda x.M \in \mathcal{D}$,

3. if $M \in \mathcal{D}$ and $N \in \mathcal{D}$, then $MN \in \mathcal{D}$.

4. if $M[x := N] \in \mathcal{D}$ and $N \in \mathcal{D}$, then $(\underline{\lambda} x.M)N \in \mathcal{D}$.

It is easy to prove that $\underline{\Lambda} = \mathcal{D}$.

Lemma 4.6. If $M \in \mathcal{D}$ and $N \in \mathcal{D}$ then $M[x := N] \in \mathcal{D}$.

Proof. By induction on $M \in \mathcal{D}$. □

Proposition 4.7. $\underline{\Lambda} = \mathcal{D}$.

Proof.

$\subset$. Let $M \in \underline{\Lambda}$. We prove by induction on M that $M \in \mathcal{D}$. We prove the case that $M = (\underline{\lambda} x.P)Q$. By induction hypothesis, $P \in \mathcal{D}$ and $Q \in \mathcal{D}$. By Lemma 4.6 we have that $P[x := Q] \in \mathcal{D}$ and by the definition of $\mathcal{D}$ we have that $(\underline{\lambda} x.P)Q \in \mathcal{D}$.

$\supset$. Let $M \in \mathcal{D}$. By induction on the derivation of $M \in \mathcal{D}$ we prove that $M \in \underline{\Lambda}$. We prove the case that $M = (\underline{\lambda} x.P)Q$. By induction hypothesis, $P[x := Q] \in \underline{\Lambda}$ and $Q \in \underline{\Lambda}$. This yields $P \in \underline{\Lambda}$. Hence $(\underline{\lambda} x.P)Q \in \underline{\Lambda}$.

□

So finiteness of developments is equivalent to the fact that each term in $\mathcal{D}$ is strongly $\underline{\beta}$-normalising. We use the following lemma of which the proof is immediate.

Lemma 4.8. If P in PQ is not of the form $\underline{\lambda} x.P_0$, then all $\underline{\beta}$-reducts of PQ are of the form $P'Q'$ with $P \twoheadrightarrow_{\underline{\beta}} P'$ and $Q \twoheadrightarrow_{\underline{\beta}} Q'$.

Theorem 4.9. If $M \in \mathcal{D}$, then all $\underline{\beta}$-rewrite sequences starting in M are finite.

Proof. The proof proceeds by induction on the derivation of $M \in \mathcal{D}$.

1. If M is a variable then it is trivial.

2. Let $M = \lambda x.P$ with $P \in \mathcal{D}$. By induction hypothesis, we have that P is strongly $\underline{\beta}$-normalising. So M is strongly $\underline{\beta}$-normalising.

3. Let $M = PQ$ with $P \in \mathcal{D}$ and $Q \in \mathcal{D}$. Note that P is not of the form $\underline{\lambda} x.P_0$. By lemma 4.8, every $\underline{\beta}$-reduct of M is of the form $P'Q'$ with $P \twoheadrightarrow_{\underline{\beta}} P'$ and $Q \twoheadrightarrow_{\underline{\beta}} Q'$. By induction hypothesis there are no infinite $\underline{\beta}$-rewrite sequences starting in P or in Q. Therefore M is strongly $\underline{\beta}$-normalising.

4. Let $M = (\underline{\lambda} x.P)Q$ with $P[x := Q] \in \mathcal{D}$ and $Q \in \mathcal{D}$. Consider an arbitrary $\underline{\beta}$-rewrite sequence $\rho : M = M_0 \twoheadrightarrow_{\underline{\beta}} M_1 \twoheadrightarrow_{\underline{\beta}} M_2 \twoheadrightarrow_{\underline{\beta}} \ldots$. There are two possibilities: in ρ the head redex of M is contracted or the head redex of M is not contracted.

   In the first case there is an i such that $M_i = P'[x := Q']$, with $P \twoheadrightarrow_{\underline{\beta}} P'$ and $Q \twoheadrightarrow_{\underline{\beta}} Q'$. The term M_i is a result of rewriting $P[x := Q]$, and the latter is by induction hypothesis strongly $\underline{\beta}$-normalising. Hence ρ is finite.

   In the second case all terms in ρ are of the form $(\underline{\lambda} x.P')Q'$ with $P \twoheadrightarrow_{\underline{\beta}} P'$ and $Q \twoheadrightarrow_{\underline{\beta}} Q'$. By induction hypothesis, $P[x := Q]$ is strongly $\underline{\beta}$-normalising, which yields that P is strongly $\underline{\beta}$-normalising, and moreover Q is strongly $\underline{\beta}$-normalising. Hence all terms in ρ are strongly normalising so ρ is finite.

□

COROLLARY 4.10. All β-developments are finite.

REMARK 4.11. It is possible to prove in a different way, also using the set $\mathcal{SN}$, that all developments are finite. We define a morphism

$$\mathcal{I} : (\underline{\Lambda},\ \{\underline{\beta}\} \times \{0,1\}^*,\ \{\stackrel{\phi}{\to}_{\underline{\beta}}\}_\phi) \to (\mathcal{SN},\ \{\beta\} \times \{0,1\}^*,\ \{\stackrel{\phi}{\to}_\beta\}_\phi)$$

Let Abs denote a distinguished variable. The definition of position has to be adapted, such that applications of Abs are not counted. Then, for instance $(\mathsf{Abs}\lambda x.M)\backslash\phi = (\lambda x.M)\backslash\phi$. We leave out the details.

First we define $\mathcal{I}_0 : \underline{\Lambda} \to \mathcal{SN}$ by induction on the definition of $\underline{\Lambda}$ as follows:

1. $\mathcal{I}(x) = x$,
2. $\mathcal{I}(\lambda x.M) = \mathsf{Abs}\lambda x.\mathcal{I}(M)$
3. $\mathcal{I}(MN) = \mathcal{I}(M)\mathcal{I}(N)$,
4. $\mathcal{I}((\underline{\lambda} x.M)N) = (\lambda x.\mathcal{I}(M))\mathcal{I}(N)$.

Next we define the mapping $\mathcal{I}_1 : \{\underline{\beta}\} \times \{0,1\}^* \to \{\beta\} \times \{0,1\}^*$ by $\mathcal{I}_1(\underline{\beta}, \phi) = (\beta, \phi)$.

We have the following:

1. if $M \in \underline{\Lambda}$ then $\mathcal{I}(M) \in \mathcal{SN}$,
2. if $M \in \underline{\Lambda}$ and $M \stackrel{\phi}{\to}_{\underline{\beta}} N$, then $\mathcal{I}(M) \stackrel{\phi}{\to}_\beta \mathcal{I}(N)$.

For the first point we need to prove that $\mathcal{I}(M[x := N]) = \mathcal{I}(M)[x := \mathcal{I}(N)]$.

5. SUPERDEVELOPMENTS

In [Raa93], *superdevelopments* were introduced and proved to be finite. Superdevelopments form an extension of the notion of development. In a superdevelopment not only redexes that descend from the initial term may be contracted, but also some redexes that are created during reduction.

There are three ways of creating new redexes (see [Lév78]):

1. $((\lambda x.\lambda y.M)N)P \to_\beta (\lambda y.M[x := N])P$
2. $(\lambda x.x)(\lambda y.M)N \to_\beta (\lambda y.M)N$
3. $(\lambda x.C[xM])(\lambda y.N) \to_\beta C'[(\lambda y.N)M']$ where C' and M' are obtained from C and M by replacing all free occurrences of x by $(\lambda y.N)$.

The first two kinds of created redexes are 'innocent' and they may be contracted in a superdevelopment. Here the redexes are created 'upwards', whereas in the last case redexes are created 'downwards'. The result that all superdevelopments are finite shows that infinite β-reduction sequences are due to the presence of the third type of redexes.

In this section we give a new proof of the fact that in λ-calculus all β-superdevelopments terminate.

First we shortly repeat the definition of a superdevelopment. The definition makes use of a set of labelled λ-terms and a notion of labelled β-reduction on it. Since application nodes will be labelled, we write them explicitly.

DEFINITION 5.1. The set Λ_l of labelled λ-terms is defined as the smallest set satisfying the following:

1. $x \in \Lambda_l$ for every variable x,
2. if $M \in \Lambda_l$ and $i \in \mathbb{N}$, then $\lambda_i x.M \in \Lambda_l$,
3. if $M, N \in \Lambda_l$ and $X \subset \mathbb{N}$, then $@^X(M, N) \in \Lambda_l$.

On the set Λ_l, the β_l-rule is defined as follows:

$$@^X(\lambda_i x.M, N) \to M[x := N] \quad \text{if } i \in X$$

DEFINITION 5.2.

1. A term $M \in \Lambda_l$ is said to be *well-labelled* if the label X of an application node never contains the label i of a λ oustide the scope of the application node.
2. A term $M \in \Lambda_l$ is *initially labelled* if it is well-labelled and all λ's have a different label.

The set of well-labelled terms is closed under β_l-reduction. In the sequel we shall suppose terms in Λ_l to be well-labelled. We define a mapping from Λ_l to Λ that erases the labels.

DEFINITION 5.3.

1. The mapping $\mathcal{E}_0 : \Lambda_l \to \Lambda$ is defined by induction on the definition of Λ_l.
   (a) $\mathcal{E}_0(x) = x$,
   (b) $\mathcal{E}_0(\lambda_i x.M) = \lambda x.\mathcal{E}_0(M)$,
   (c) $\mathcal{E}_0(@^X(M, N)) = \mathcal{E}_0(M)\mathcal{E}_0(N)$.
2. The mapping $\mathcal{E}_1 : \{\beta_l\} \times \{0,1\}^* \to \{\beta\} \times \{0,1\}^*$ is defined by $\mathcal{E}_1(\beta_l, \phi) = (\beta, \phi)$.

The proof of the following lemma is straightforward.

LEMMA 5.4. The mapping $\mathcal{E} = (\mathcal{E}_0, \mathcal{E}_1)$ is a morphism from $(\Lambda_l,\ \{\beta_l\} \times \{0,1\}^*,\ \{\stackrel{\phi}{\to}_{\beta_l}\}_\phi)$ to $(\Lambda,\ \{\beta\} \times \{0,1\}^*,\ \{\stackrel{\phi}{\to}_{\beta}\}_\phi)$.

The definition of a superdevelopment is as follows.

DEFINITION 5.5. A rewrite sequence $\sigma : M \twoheadrightarrow_\beta N$ in Λ is a *superdevelopment* if there is a rewrite sequence ρ in Λ_l that starts in an initially labelled term and that is an $\mathcal{E}$-lifting of σ.

We give a new proof of the fact that all superdevelopments are finite. It is similar to the proof of finite developments in section 4.

DEFINITION 5.6. The set $\mathcal{SD}$ is the smallest subset of the set of lambda terms satisfying

1. $x \in \mathcal{SD}$ for all variables x,

2. if $M \in \mathcal{SD}$, then $\lambda x.M \in \mathcal{SD}$,
3. if $M \in \mathcal{SD}$ and $N \in \mathcal{SD}$, then $MN \in \mathcal{SD}$,
4. if $M[x := N]P_1 \dots P_n \in \mathcal{SD}$ and $N \in \mathcal{SD}$, then $(\underline{\lambda} x.M)NP_1 \dots P_n \in \mathcal{SD}$,
5. if $(\underline{\lambda} y.M)NP_1 \dots P_n \in \mathcal{SD}$, then $(\underline{\lambda} x.x)(\underline{\lambda} y.M)NP_1 \dots P_n \in \mathcal{SD}$.

The $\underline{\beta}$-rewrite relation is defined as follows:

$$(\underline{\lambda} x.M)N \to_{\underline{\beta}} M[x := N]$$

We need to prove that a rewrite sequence $M \twoheadrightarrow_\beta N$ can be lifted to a β_l-rewrite sequence starting in M with an initial labelling if and only if it can be lifted to a $\underline{\beta}$-rewrite sequence starting with $M \in \mathcal{SD}$. To prove this is routine. Then, proving that all superdevelopments are finite is equivalent to proving that all $\underline{\beta}$-rewrite sequences in $\mathcal{SD}$ are finite. The latter is proved by a trivial induction on the set $\mathcal{SD}$ using the following two lemma's.

LEMMA 5.7. **($\underline{\beta}$-Closure)** Let $M \in \mathcal{SD}$. If $M \to_{\underline{\beta}} M'$ then $M' \in \mathcal{SD}$.

LEMMA 5.8. Let $M = PQ$ with $P \in \mathcal{SD}$ and $Q \in \mathcal{SD}$. If $M \twoheadrightarrow_{\underline{\beta}} M'$, then $M' = P'Q'$ with $P \twoheadrightarrow_{\underline{\beta}} P'$ and $Q \twoheadrightarrow_{\underline{\beta}} Q'$.

THEOREM 5.9. If $M \in \mathcal{SD}$, then all $\underline{\beta}$-rewrite sequences starting at M are finite.

Proof. The proof proceeds by induction on the derivation of $M \in \mathcal{SD}$.

1. If M is a variable then it is trivial.
2. Let $M = \lambda x.P$ with $P \in \mathcal{SD}$. By induction hypothesis, we have that P is strongly $\underline{\beta}$-normalising. So M is strongly $\underline{\beta}$-normalising.
3. Let $M = PQ$ with $P \in \mathcal{SD}$ and $Q \in \mathcal{SD}$. By induction hypothesis P and Q are $\underline{\beta}$-strongly normalising. It follows from lemma 5.8 that any $\underline{\beta}$-sequence starting at M is finite.
4. Let $M = (\underline{\lambda} x.P)QN_1 \dots N_n$ with $P[x := Q]N_1 \dots N_n \in \mathcal{SD}$. Consider an arbitrary $\underline{\beta}$-rewrite sequence $\rho : M = M_0 \twoheadrightarrow_{\underline{\beta}} M_1 \twoheadrightarrow_{\underline{\beta}} M_2 \twoheadrightarrow_{\underline{\beta}} \dots$. There are two possibilities: in ρ the head redex of M is contracted or the head redex of M is not contracted.

   In the first case, there is an i such that $M_i = P'[x := Q']N'_1 \dots N'_n$ with $P \twoheadrightarrow_{\underline{\beta}} P'$, $Q \twoheadrightarrow_{\underline{\beta}} Q', N_1 \twoheadrightarrow_{\underline{\beta}} N'_1, \dots, N_n \twoheadrightarrow_{\underline{\beta}} N'_n$. The term M_i is obtained by rewriting $P[x := Q]N_1 \dots N_n$ and the latter term is by induction hypothesis strongly $\underline{\beta}$-normalising. Hence ρ is finite.

   In the second case, all terms in ρ are of the form $(\underline{\lambda} x.P')Q'N'_1 \dots N'_n$ with $P \twoheadrightarrow_{\underline{\beta}} P', Q \twoheadrightarrow_{\underline{\beta}} Q', N_1 \twoheadrightarrow_{\underline{\beta}} N'_1, \dots, N_n \twoheadrightarrow_{\underline{\beta}} N'_n$. Since $P[x := Q]N_1 \dots N_n$ and Q are by induction hypothesis strongly $\underline{\beta}$-normalising, we have that $P, Q, N_1, \dots, N_n$ are strongly $\underline{\beta}$-normalising. So all terms in ρ are strongly $\underline{\beta}$-normalising and hence ρ is finite.

5. Let $M = (\underline{\lambda}x.x)(\underline{\lambda}y.N)PN_1 \ldots N_n$ with $(\underline{\lambda}y.N)PN_1 \ldots N_n \in \mathcal{SD}$. Consider an arbitrary $\underline{\beta}$-rewrite sequence $\rho : M = M_0 \twoheadrightarrow_{\underline{\beta}} M_1 \twoheadrightarrow_{\underline{\beta}} M_2 \twoheadrightarrow_{\underline{\beta}} \ldots$. There are two possibilities: in ρ the head redex of M is contracted or the head redex of M is not contracted.

   In the first case, there is an i such that $M_i = (\underline{\lambda}y.N')P'N'_1 \ldots N'_n$ with $N \twoheadrightarrow_{\underline{\beta}} N', P \twoheadrightarrow_{\underline{\beta}} P', N_1 \twoheadrightarrow_{\underline{\beta}} N'_1, \ldots, N_n \twoheadrightarrow_{\underline{\beta}} N'_n$. The term M_i is obtained by rewriting the term $(\underline{\lambda}y.N)PN_1 \ldots N_n$ and the latter term is by induction hypothesis strongly $\underline{\beta}$-normalising. So M_i is strongly $\underline{\beta}$-normalising and hence ρ is finite.

   In the second case, all terms in ρ are of the form $(\underline{\lambda}x.x)(\underline{\lambda}y.N')P'N'_1 \ldots N'_n$ with $N \twoheadrightarrow_{\underline{\beta}} N', P \twoheadrightarrow_{\underline{\beta}} P', N_1 \twoheadrightarrow_{\underline{\beta}} N'_1, \ldots, N_n \twoheadrightarrow_{\underline{\beta}} N'_n$. By induction hypothesis, $(\underline{\lambda}y.N)PN_1 \ldots N_n$ is strongly $\underline{\beta}$-normalising. Hence $N, P, N_1, \ldots, N_n$ are all strongly $\underline{\beta}$-normalising. This yields that ρ is finite.

□

REMARK 5.10. Another proof of the fact that all superdevelopments are finite can be given in a way similar to the one in Remark 5.9. That is, we define a morphism from $(\Lambda_l,\ \{\beta_l\} \times \{0,1\}^*,\ \{\xrightarrow{\phi}_{\beta_l}\}_\phi)$ to $(\mathcal{SN},\ \{\beta\} \times \{0,1\}^*,\ \{\xrightarrow{\phi}_{\beta}\}_\phi)$, using the following function.

DEFINITION 5.11. The function $|-| : \Lambda_l \to \Lambda_l$ is defined by induction on the definition of Λ_l.

1. $|x| = x$,
2. $|\lambda_i x.M| = \lambda_i x.|M|$,
3. $|@^X(M,N)| = \begin{cases} M_0[x := |N|] & \text{if } |M| = \lambda_i x.M_0 \text{ and } i \in X \\ @^X(|M|,|N|) & \text{otherwise} \end{cases}$

Let App denote a distinguished variable. Like in the case of developments, the definition of position has to be adapted, such that occurrences of App do not count. Then, for instance $(\mathsf{App}MN)\backslash\phi = (MN)\backslash\phi$. As in the case of developments, we leave out the details. Now we can define the morphism $\mathcal{J} : (\Lambda_l,\ \{\beta_l\} \times \{0,1\}^*,\ \{\xrightarrow{\phi}_{\beta_l}\}_\phi) \to (\mathcal{SN},\ \{\beta\} \times \{0,1\}^*,\ \{\xrightarrow{\phi}_{\beta}\}_\phi)$.

DEFINITION 5.12.

1. The mapping $\mathcal{J}_0 : \Lambda_l \to \mathcal{SN}$ is defined by induction on the definition of $\underline{\Lambda}$.
   (a) $\mathcal{J}_0(x) = x$,
   (b) $\mathcal{J}_0(\lambda_i x.M) = \lambda x.\mathcal{J}_0(M)$,
   (c) $\mathcal{J}_0(@^X(M,N)) = \begin{cases} \mathcal{J}_0(M)\mathcal{J}_0(N) & \text{if } |M| = \lambda_i x.M_0 \text{ and } i \in X \\ \mathsf{App}\mathcal{J}_0(M)\mathcal{J}_0(N) & \text{otherwise} \end{cases}$
2. The mapping $\mathcal{J}_1 : \{\beta_l\} \times \{0,1\}^* \to \{\beta\} \times \{0,1\}^*$ is defined by $\mathcal{E}_1(\beta_l,\phi) = (\beta,\phi)$.

LEMMA 5.13. Let $M \in \Lambda_l$, $\mathcal{J}(M) \in \mathcal{SN}$ and $N \in \mathcal{SN}$. Then $\mathcal{J}(M)[x := N] \in \mathcal{SN}$.

Proof. The proof proceeds by induction on the derivation of $\mathcal{J}(M) \in \mathcal{SN}$.

1. Suppose $\mathcal{J}(M) = yP_1 \dots P_n$ with $P_i \in \mathcal{SN}$ for $i = 1, \dots, n$. Note that it must be the case that $y = \mathsf{App}$. By induction hypothesis, $\mathcal{J}(P_i)[x := N] \in \mathcal{SN}$ for $i = 1, \dots, n$. Hence $\mathcal{J}(M)[x := N] \in \mathcal{SN}$.

2. Suppose $\mathcal{J}(M) = \lambda y.P$ with $P \in \mathcal{SN}$. Using induction hypothesis we obtain that $\mathcal{J}(M)[x := N] \in \mathcal{SN}$.

3. Suppose $\mathcal{J}(M) = (\lambda y.P)Q_1Q_2 \dots Q_n$ with $P[y := Q_1]Q_2 \dots Q_n \in \mathcal{SN}$ and $Q_1 \in \mathcal{SN}$. By induction hypothesis, we have $(P[y := Q_1]Q_2 \dots Q_n)[x := N] \in \mathcal{SN}$ and $Q_1[x := N] \in \mathcal{SN}$. This yields $\mathcal{J}(M)[x := N] \in \mathcal{SN}$.

□

Lemma 5.14. Let $M \in \Lambda_l$, $\mathcal{J}(M) \in \mathcal{SN}$ and $N \in \mathcal{SN}$. Then $\mathcal{J}(M)N \in \mathcal{SN}$.

Proof. The proof proceeds by induction on the derivation of $\mathcal{J}(M) \in \mathcal{SN}$.

1. Suppose $\mathcal{J} = xP_1 \dots P_n$ with $P_i \in \mathcal{SN}$ for $i = 1, \dots, n$. Then $\mathcal{J}(M)N \in \mathcal{SN}$.

2. Suppose $\mathcal{J}(M) = \lambda x.P$ with $P \in \mathcal{SN}$. Then $M = \lambda_i x.M_0$ and $\mathcal{J}(M_0) = P$. By the previous lemma we have $P[x := N] \in \mathcal{SN}$. Hence $\mathcal{J}(M)N \in \mathcal{SN}$.

3. Suppose $\mathcal{J}(M) = (\lambda x.P)Q_1Q_2 \dots Q_n$ with $P[x := Q_1]Q_2 \dots Q_n \in \mathcal{SN}$ and $Q_1 \in \mathcal{SN}$. By induction hypothesis, we have $P[x := Q_1]Q_2 \dots Q_nN \in \mathcal{SN}$. Moreover $Q_1 \in \mathcal{SN}$, hence $\mathcal{J}(M)N \in \mathcal{SN}$.

□

Theorem 5.15. Let $M \in \Lambda_l$. Then $\mathcal{J}(M) \in \mathcal{SN}$.

Proof. The proof proceeds by induction on $M \in \Lambda_l$ and makes use of the two previous lemmas. □

Theorem 5.16. Let $M \in \Lambda_l$. If $M \stackrel{\phi}{\to}_{\beta_l} N$ in Λ_l then $\mathcal{J}(M) \stackrel{\phi}{\to}_{\beta} \mathcal{J}(N)$ in $\mathcal{SN}$.

It follows from Theorem 5.15 and Theorem 5.16 that $\mathcal{J}$ is a morphism from $(\Lambda_l, \{\stackrel{\phi}{\to}_{\beta_l}\}_\phi)$ to $(\mathcal{SN}, \{\stackrel{\phi}{\to}_{\beta}\}_\phi)$. As an immediate consequence of this, we have that all superdevelopments are finite.

Finally we would like to remark that it is easy to prove the following lemma.

Lemma 5.17. Let $M \in \mathcal{SD}$.

1. $M \twoheadrightarrow_{\underline{\beta}} |M|$.

2. If $M \to_{\underline{\beta}} M'$ then $|M| = |M'|$.

So we have $|M|$ is the β_l-normal form of M and it is unique. As a consequence of this we have that β_l is Church-Rosser.

6. Two Perpetual Strategies

In this section we consider two rewrite strategies, F_{bk} defined in [BK82] and F_∞ introduced in [BBKV76]. Both strategies are *perpetual*, which means that they yield an infinite rewrite sequence whenever possible. We give for both strategies a new proof of the fact that they are perpetual. Our proofs are simpler than the original ones and make in both cases use of the characterisation of strongly normalising terms.

For the sake of self-containment we first give some definitions that can for instance be found in [Bar84].

Perpetual strategies.

Definition 6.1.

1. A *strategy for β-reduction* is a map $F : \Lambda \to \Lambda$ such that for all $M \in \Lambda$, $M \twoheadrightarrow_\beta F(M)$.
2. A *one-step strategy for β-reduction* is a map $F : \Lambda \to \Lambda$ such that for all $M \in \Lambda$ not in β-normal form, $M \to_\beta F(M)$.

Definition 6.2. A strategy is called *perpetual* if $F(M)$ is strongly normalising implies M is strongly normalising.

A perpetual strategy finds an infinite rewrite sequence if possible. Perpetual strategies are interesting because of the easy observation that a term M is strongly normalising if and only if a perpetual strategy finds a finite rewrite sequence starting from M.

In the sequel we will deal with one-step strategies only.

Definition 6.3. Let F be a strategy for β-reduction. An *F-rewrite sequence of M* is defined as

$$M \to_\beta F(M) \to_\beta F^2(M) \to_\beta \dots$$

possibly ending in the normal form of M.

The strategy F_{bk}. First we consider the strategy F_{bk} as introduced in [BK82]. We give a simple proof that F_{bk} is perpetual using Definition 3.1.

Definition 6.4. Suppose that $M \in \Lambda$ is not in normal form.
Let $M = C[(\lambda x.P)Q]$ where $(\lambda x.P)Q$ is the leftmost redex of M.

$$F_{bk}(C[(\lambda x.P)Q]) = \begin{cases} C[P[x := Q]] & \text{if } Q \text{ is strongly normalising} \\ C[(\lambda x.P)F_{bk}(Q)] & \text{otherwise} \end{cases}$$

Note that the the strategy F_{bk} yields standard rewrite sequences.

Theorem 6.5. F_{bk} is a perpetual strategy.

Proof. Suppose that M is not in normal form and $F_{bk}(M)$ is strongly normalising. Then $F_{bk}(M) \in \mathcal{SN}$. We prove $M \in \mathcal{SN}$, which is equivalent to M is strongly normalising. We prove $M \in \mathcal{SN}$ by induction on the number of steps in the derivation of $F_{bk}(M) \in \mathcal{SN}$.
The term M is of the form $\lambda x_1 \dots x_n.PQ_1 \dots Q_m$ where P can be either a variable y or an abstraction $\lambda y.P_0$. We consider these two cases:

1. $P = y$. Then

$$F_{bk}(M) = \lambda x_1 \ldots \lambda x_n.y\ Q_1 \ldots Q_{i-1}\ F_{bk}(Q_i) Q_{i+1} \ldots Q_m$$

   where $Q_1, \ldots Q_{i-1}$ are in normal form.

   Since $F_{bk}(M) \in \mathcal{SN}$ we have that $F_{bk}(Q_i) \in \mathcal{SN}$. It follows from the induction hypothesis that $Q_i \in \mathcal{SN}$. This yields $M = \lambda x_1 \ldots \lambda x_n.y Q_1 \ldots Q_m \in \mathcal{SN}$.

2. $P = \lambda y.P_0$. Suppose Q_1 is not strongly normalising. Then $Q_1 \notin \mathcal{SN}$. This yields a contradiction with the hypothesis $F_{bk}(M) \in \mathcal{SN}$. Hence $Q_1 \in \mathcal{SN}$ and

$$F_{bk}(M) = \lambda x_1 \ldots \lambda x_n.P_0[y := Q_1] Q_2 \ldots Q_m$$

   Now $F_{bk}(M) \in \mathcal{SN}$ and $Q_1 \in \mathcal{SN}$ yield that $M \in \mathcal{SN}$.

□

The strategy F_∞. We now consider the strategy F_∞ that is defined in [BBKV76]. This strategy does not check whether the argument of the left-most redex is strongly normalising or not. Instead, it is checked whether the left-most redex is an I-redex. If it is, it is contracted. If it is not, contracting it could imply loosing the possibility of having an infinite reduction sequence. Therefore, in that case, the left-most redex is only contracted if the argument is a normal form. If the argument is not a normal form, the strategy is applied to the argument.

Definition 6.6. Suppose that $M \in \Lambda$ is not in normal form.
Let $M = C[(\lambda x.P)Q]$ where $(\lambda x.P)Q$ is the leftmost redex of M.

$$F_\infty(C[(\lambda x.P)Q]) = \begin{cases} C[(\lambda x.P)F_\infty(Q)] & \text{if } x \notin \mathsf{FV}(P) \text{ and } Q \notin \mathcal{NF} \\ C[P[x := Q]] & \text{otherwise} \end{cases}$$

The F_∞-rewrite sequence of a term is not necessarily a standard rewrite sequence. The merit of F_∞, however, is that it is decidable. We prove that F_∞ is perpetual. Note that this proof is simpler than the proof in [BBKV76] or in [Bar84] (Chapter IV paragraph 4).

Theorem 6.7. F_∞ is a perpetual strategy.

Proof. Suppose that M is not in normal form and $F_\infty(M)$ is strongly normalising. Then $F_\infty(M) \in \mathcal{SN}$. We prove that $M \in \mathcal{SN}$ which means that M is strongly normalising. The proof proceeds by induction on the derivation of $F_\infty(M) \in \mathcal{SN}$.

The term M is of the form $\lambda x_1 \ldots \lambda x_n.PQ_1 \ldots Q_m$ where P can be either a variable y or an abstraction $\lambda y.P_0$. We consider these two cases:

1. $P = y$. Then

$$F_\infty(M) = \lambda x_1 \ldots \lambda x_n.y\ Q_1 \ldots Q_{i-1} F_\infty(Q_i) Q_{i+1} \ldots Q_m$$

   where $Q_1, \ldots Q_{i-1}$ are in normal form.

   Since $F_\infty(M) \in \mathcal{SN}$ we have that $F_\infty(Q_i) \in \mathcal{SN}$. It follows from the induction hypothesis that $Q_i \in \mathcal{SN}$. This yields $M = \lambda x_1 \ldots \lambda x_n.y Q_1 \ldots Q_m \in \mathcal{SN}$.

2. $P = \lambda y.P_0$. Two cases are distinguished.

   (a) $y \in \mathsf{FV}(P_0)$. Then

$$F_\infty(M) = \lambda x_1 \dots \lambda x_n.P_0[y := Q_1]Q_2 \dots Q_m \in \mathcal{SN}$$

   Since $y \in P_0$ we have that $Q_1 \in \mathcal{SN}$. By the definition of the set $\mathcal{SN}$ we have that $M \in \mathcal{SN}$.

   (b) $y \notin \mathsf{FV}(P_0)$.
   If Q_1 is a normal form, then

$$F_\infty(M) = \lambda x_1 \dots \lambda x_n.P_0 Q_2 \dots Q_m$$

   Since $F_\infty(M) \in \mathcal{SN}$, that is, $\lambda x_1 \dots \lambda x_n.P_0[y := Q_1]Q_2 \dots Q_m \in \mathcal{SN}$, and moreover clearly $Q_1 \in \mathcal{SN}$, we can conclude $M = \lambda x_1 \dots \lambda x_n.(\lambda y.P_0)Q_1 Q_2 \dots Q_m \in \mathcal{SN}$.
   If Q_1 is not a normal form, then

$$F_\infty(M) = \lambda x_1 \dots \lambda x_n.(\lambda y.P_0)F_\infty(Q_1)\ Q_2 \dots Q_m$$

   Since $F_\infty(M) \in \mathcal{SN}$ we have $F_\infty(Q_1) \in \mathcal{SN}$ and $P_0\ Q_2 \dots Q_m \in \mathcal{SN}$. By induction hypothesis we have that $Q_1 \in \mathcal{SN}$.
   We apply the last clause of Definition 3.1 in order to obtain $(\lambda y.P_0)Q_1\ Q_2 \dots Q_m \in \mathcal{SN}$. We have $M = \lambda x_1 \dots x_n.(\lambda y.P_0)Q_1\ Q_2 \dots Q_m \in \mathcal{SN}$ by applying n times the second clause of Definition 3.1.

□

7. A Maximal Strategy

In this section we prove that the strategy F_∞ is *maximal*, which means that it computes for each term M the longest possible rewrite sequence. In particular, a maximal strategy is perpetual. The converse is not necessarily true, as witnessed by the strategy F_{bk} defined in [BK82].

Our proof that F_∞ is a maximal strategy makes use of the characterisation of strongly normalising terms. We define a mapping h that computes the length of a F_∞-rewrite sequence of a term. Then it is proved that the mapping h computes the maximal length of a reduction to normal form.

We start by giving some definitions.

Definition 7.1. Let σ be a rewrite sequence. The *length* of σ, denoted by $\|\sigma\|$, is the number of rewrite steps in σ. We have that $\|\sigma\|$ is either a natural number or ∞.

Definition 7.2. A rewrite sequence $\sigma : M \twoheadrightarrow N$ is *maximal* if for all $\rho : M \twoheadrightarrow N$ we have $\|\sigma\| \geq \|\rho\|$.

Definition 7.3. A strategy F is *maximal* for each term M the F-rewrite sequence of M is maximal.

We define a map $h : \Lambda \to \mathbb{N} \cup \{\infty\}$ that computes for each term the length of its F_∞-rewrite sequence.

Definition 7.4.

1. The map $h : \mathcal{SN} \to \mathbb{N}$ is defined by induction on the definition of $\mathcal{SN}$.

$$h(xM_1 \dots M_n) = \begin{cases} 0 & \text{if } n = 0 \\ \sum_{i=1}^{n} h(M_i) & \text{if } n \neq 0 \end{cases}$$

$$h(\lambda x.M) = h(M)$$

$$h((\lambda x.M)NP_1 \dots P_n) = \begin{cases} h(M[x := N]P_1 \dots P_n) + 1 & \text{if } x \in M \\ h(M\ P_1 \dots P_n) + h(N) + 1 & \text{if } x \notin M \end{cases}$$

2. We extend $h : \mathcal{SN} \to \mathbb{N}$ to $h : \Lambda \to \mathbb{N} \cup \{\infty\}$ by defining $h(M) = \infty$ if $M \notin \mathcal{SN}$.

We prove that the map h has the following two properties:

- it computes the length of the F_∞-rewrite sequence of a term M,
- it computes the maximum length of all rewrite sequences starting in M.

From these we will conclude that F_∞ is a maximal strategy.
First we prove the following lemma.

Lemma 7.5. Let $M \in \mathcal{SN}$.

1. If $M \in \mathcal{NF}$ then $h(M) = 0$.
2. If $M \notin \mathcal{NF}$ then $h(M) = h(F_\infty(M)) + 1$.

Proof.

1. Trivial.
2. Suppose that M is not in normal form. We prove that $h(M) = h(F_\infty(M)) + 1$ by induction on the number of steps in the derivation of $M \in \mathcal{SN}$.
   The term M is of the form $\lambda x_1 \dots \lambda x_n.PQ_1 \dots Q_m$ where P can be either a variable y or an abstraction $\lambda y.P_0$. We consider these two cases:

   (a) $P = y$. Then

$$F_\infty(M) = \lambda x_1 \dots \lambda x_n.y\ Q_1 \dots Q_{i-1} F_\infty(Q_i) Q_{i+1} \dots Q_m$$

where $Q_1, \ldots Q_{i-1}$ are in normal form. By induction hypothesis we have $h(Q_i) = h(F_\infty(Q_i)) + 1$. Hence we have

$$\begin{aligned} h(M) &= h(\lambda x_1 \ldots \lambda x_n . y Q_1 \ldots Q_m) \\ &= \sum_{k=i}^{m} h(Q_k) \\ &= h(Q_i) + \sum_{k=i+1}^{m} h(Q_k) \\ &= h(F_\infty(Q_i)) + 1 + \sum_{k=i+1}^{m} h(Q_k) \\ &= h(F_\infty(M)) + 1 \end{aligned}$$

(b) $P = (\lambda y . P_0)$. Two cases are distinguished.

i. $y \in \mathsf{FV}(P_0)$. Then

$$F_\infty(M) = \lambda x_1 \ldots \lambda x_n . P_0[y := Q_1]\ Q_2 \ldots Q_m$$

We have

$$\begin{aligned} h(M) &= h(\lambda x_1 \ldots \lambda x_n . (\lambda y . P_0) Q_1 Q_2 \ldots Q_m) \\ &= h(P_0[y := Q_1]\ Q_2 \ldots Q_m) + 1 \\ &= h(F_\infty(M)) + 1 \end{aligned}$$

ii. $y \notin \mathsf{FV}(P_0)$. Again two cases are distinguished.

A. If Q_1 is not in normal form then

$$F_\infty(M) = \lambda x_1 \ldots \lambda x_n . (\lambda y . P_0) F_\infty(Q_1)\ Q_2 \ldots Q_m$$

By induction hypothesis, $h(Q_1) = h(F_\infty(Q_1)) + 1$. Hence we have

$$\begin{aligned} h(M) &= h(\lambda x_1 \ldots \lambda x_n . (\lambda y . P_0) Q_1 Q_2 \ldots Q_m) \\ &= h(P_0\ Q_2 \ldots Q_m) + h(Q_1) + 1 \\ &= h(P_0\ Q_2 \ldots Q_m) + h(F_\infty(Q_1)) + 1 + 1 \\ &= h(F_\infty(M)) + 1 \end{aligned}$$

B. If Q_1 is in normal form then

$$F_\infty(M) = \lambda x_1 \ldots \lambda x_n . P_0\ Q_2 \ldots Q_m$$

We have

$$\begin{aligned} h(M) &= h(\lambda x_1 \ldots \lambda x_n . (\lambda y . P_0) Q_1 Q_2 \ldots Q_m) \\ &= h(P_0\ Q_2 \ldots Q_m) + h(Q_1) + 1 \\ &= h(P_0\ Q_2 \ldots Q_m) + 0 + 1 \\ &= h(F_\infty(M)) + 1 \end{aligned}$$

□

THEOREM 7.6. The map $h : \Lambda \to \mathbb{N} \cup \{\infty\}$ computes the length of the F_∞-rewrite sequence of a term M.

Proof. If $M \in \mathcal{SN}$ then $M \to_\beta F_\infty(M) \to_\beta \ldots\ldots \to_\beta F^n_\infty(M) = \mathsf{nf}(M)$. It follows by induction on n that $h(M) = n$ using Lemma 7.5.
If $M \notin \mathcal{SN}$ then the F_∞-rewrite sequence of M is infinite and indeed $h(M) = \infty$. □

Now we prove that $h : \Lambda \to \mathbb{N} \cup \{\infty\}$ computes the maximum length of all reductions sequences starting at M. Here $\mathsf{maxred}(M)$ denotes the length of a maximal rewrite sequence starting in M.

THEOREM 7.7. Let $M \in \Lambda$. We have

$$h(M) = \mathsf{maxred}(M)$$

Proof. If $M \notin \mathcal{SN}$, then $h(M) = \infty$ so it is clear that the statement holds.

Suppose that $M \in \mathcal{SN}$ is not in normal form. We will prove that the length of an arbitrary reduction to normal form is less than or equal to $h(M)$. The proof proceeds by induction on the number of steps in the derivation of $M \in \mathcal{SN}$. The term M is of the form $\lambda x_1 \ldots x_n.PQ_1 \ldots Q_m$ where P can be either a variable y or an abstraction $\lambda y.P_0$. We consider these two cases:

1. $P = y$. An arbitrary reduction from M to normal form can be transformed into a reduction sequence of the same length such that:

$$\begin{array}{lll} \lambda x_1 \ldots \lambda x_n.yQ_1 \ldots Q_m & \overset{n_1}{\twoheadrightarrow}_\beta & \lambda x_1 \ldots \lambda x_n.y\ \mathsf{nf}(Q_1)Q_2 \ldots Q_m \\ & \overset{n_2}{\twoheadrightarrow}_\beta & \lambda x_1 \ldots \lambda x_n.y\ \mathsf{nf}(Q_1)\ \mathsf{nf}(Q_2) \ldots (Q_m) \\ & \twoheadrightarrow_\beta & \ldots \\ & \overset{n_m}{\twoheadrightarrow}_\beta & \lambda x_1 \ldots \lambda x_n.y\ \mathsf{nf}(Q_1)\ \mathsf{nf}(Q_2) \ldots \mathsf{nf}(Q_m) \end{array}$$

   The number of steps of this sequence is $n_1 + \ldots + n_m$. By induction hypothesis, we have $h(Q_i) \geq n_i$ for $i = 1, \ldots, m$. Hence we have

$$\begin{array}{lll} h(M) & = & \sum_{i=1}^{m} h(Q_i) \\ & \geq & \sum_{i=1}^{m} n_i \end{array}$$

2. $P = \lambda y.P_0$. Two cases are distinguished.

   (a) $y \in \mathsf{FV}(P_0)$. An arbitrary reduction sequence from M to normal form is of the form

$$\begin{array}{lll} M & = & \lambda x_1 \ldots \lambda x_n.(\lambda y.P_0)Q_1Q_2 \ldots Q_m \\ & \overset{p}{\twoheadrightarrow}_\beta & \lambda x_1 \ldots \lambda x_n.(\lambda y.P_0')Q_1'Q_2' \ldots Q_m' \\ & \to_\beta & \lambda x_1 \ldots \lambda x_n.P_0'[y := Q_1']Q_2' \ldots Q_m' \\ & \overset{l}{\twoheadrightarrow}_\beta & \mathsf{nf}(M) \end{array}$$

It can be transformed into a rewrite sequence of the form

$$\begin{array}{rcl} M & = & \lambda x_1 \dots \lambda x_n.(\lambda y.P_0)Q_1 Q_2 \dots Q_m \\ & \to_\beta & \lambda x_1 \dots \lambda x_n.P_0[y := Q_1]Q_2 \dots Q_m \\ & \overset{k}{\twoheadrightarrow}_\beta & \lambda x_1 \dots \lambda x_n.P_0'[y := Q_1']Q_2' \dots Q_m' \\ & \overset{l}{\twoheadrightarrow}_\beta & \mathsf{nf}(M) \end{array}$$

with $k \geq p$. By induction hypothesis, $h(P_0[y := Q_1]Q_2 \dots Q_m) \geq k + l$. Hence

$$\begin{array}{rcl} h(M) & = & h(P_0[y := Q_1]Q_2 \dots Q_m) + 1 \\ & \geq & k + l + 1 \\ & \geq & p + l + 1 \end{array}$$

(b) $y \notin P_0$. An arbitrary reduction sequence from M to normal form can be transformed into a reduction sequence of the same length of the form:

$$\begin{array}{rcl} M & = & \lambda x_1 \dots \lambda x_n.(\lambda y.P_0)Q_1 Q_2 \dots Q_m \\ & \overset{p}{\twoheadrightarrow}_\beta & \lambda x_1 \dots \lambda x_n.(\lambda y.P_0)Q_1' Q_2 \dots Q_m \\ & \to_\beta & \lambda x_1 \dots \lambda x_n.P_0 \ Q_2 \dots Q_m \\ & \overset{l}{\twoheadrightarrow}_\beta & \mathsf{nf}(M) \end{array}$$

By induction hypothesis we have that $h(Q_1) \geq p$ and
$h(P_0 Q_2 \dots Q_m) \geq l$. Hence

$$\begin{array}{rcl} h(M) & = & h(P_0 Q_2 \dots Q_m) + h(Q_1) + 1 \\ & \geq & l + p + 1 \end{array}$$

□

THEOREM 7.8. The strategy F_∞ is maximal.

Proof.

1. By Theorem 7.6 we have that $h(M)$ is the length of the F_∞-rewrite sequence of M.

2. By Theorem 7.7 we have that $h(M) = \mathsf{maxred}(M)$ is the maximum length of all reductions sequences starting at M.

Hence the strategy F_∞ is maximal. □

We can also define a map $h' : \Lambda \to \mathbb{N} \cup \{\infty\}$ that computes the length from M to normal form using the strategy F_{bk}.

DEFINITION 7.9.

1. The map $h' : \mathcal{SN} \to \mathbb{N}$ is defined by induction on the definition of $\mathcal{SN}$.

$$h'(x\ M_1 \dots M_n) = \sum_{i=1}^{i=n} h'(M_i)$$

$$h'(\lambda x.M) = h'(M)$$

$$h'((\lambda x.M)NP_1 \dots P_n) = h'(M[x := N]P_1 \dots P_n) + 1$$

2. The map $h' : \mathcal{SN} \to \mathbb{N}$ is extended to $h' : \Lambda \to \mathbb{N} \cup \{\infty\}$ by defining $h'(M) = \infty$ if $M \notin \mathcal{SN}$.

Lemma 7.10. The map $h' : \Lambda \to \mathbb{N} \cup \{\infty\}$ computes the length of the F_{bk}-rewrite sequence of a term M.

Note that $h'(M) \leq h(M)$ for any term M.

8. Normalisation of Simply Typed λ-calculus

In this section we give a new proof of the fact that the simply typed λ-calculus is β-strongly normalising.

In the proof we make use of the characterisation of the strongly normalising (untyped) λ-terms. We do not need to consider an interpretation for simply typed λ-terms.

First we shortly repeat the definitions of simply typed λ-calculus à la Curry.

Types, written as $\alpha, \beta, \gamma, \dots$, are built from type variables and the type constructor $\to$.

Definition 8.1. The simply typed lambda calculus $\lambda^{\to}$-Curry (or just $\lambda^{\to}$) is defined by the notion of type derivation $\Gamma \vdash_{\lambda_{\to}} M : \alpha$ (or just $\Gamma \vdash M : \alpha$) given by the following rules:

Start $\quad \Gamma \vdash x : \alpha$ if $x : \alpha \in \Gamma$

$\to$-Introduction $$\frac{\Gamma, x : \alpha \vdash M : \beta}{\Gamma \vdash \lambda x.M : \alpha \to \beta}$$

$\to$-Elimination $$\frac{\Gamma \vdash M : \alpha \to \beta \quad \Gamma \vdash N : \alpha}{\Gamma \vdash MN : \beta}$$

If A and B are subsets of Λ, then we define

$$A \to B = \{M \in \Lambda \mid \forall N \in A : MN \in B\}$$

Note that if $A \subset A'$ then $A' \to B \subset A \to B$ and that if $B \subset B'$ then $A \to B \subset A \to B'$.

We wish to consider the set of terms that are typable by a type α in a context Γ.

Definition 8.2. $\mathcal{T}(\Gamma; \alpha) = \{M \in \Lambda \mid \Gamma \vdash M : \alpha\}$.

Lemma 8.3. Let $\mathcal{T}(\Gamma; \alpha) \neq \emptyset$. Then $\mathcal{T}(\Gamma; \alpha \to \beta) = \mathcal{T}(\Gamma; \alpha) \to \mathcal{T}(\Gamma; \beta)$.

Proof.

$\subset$. Let $M \in \mathcal{T}(\Gamma; \alpha \to \beta)$ and $N \in \mathcal{T}(\Gamma; \alpha)$. Clearly $\Gamma \vdash MN : \beta$.

$\supset$. Let $M \in \mathcal{T}(\Gamma;\alpha) \to \mathcal{T}(\Gamma;\beta)$. Since $\mathcal{T}(\Gamma,\alpha) \neq \emptyset$, there is some $N \in \mathcal{T}(\Gamma,\alpha)$. Hence $MN \in \mathcal{T}(\Gamma;\beta)$. This yields $\Gamma \vdash M : \alpha \to \beta$, so $M \in \mathcal{T}(\Gamma;\alpha \to \beta)$. □

Definition 8.4. The set $\mathcal{SN}(\Gamma;\alpha)$ is defined as follows.

$$\mathcal{SN}(\Gamma;\alpha) = \{M \in \mathcal{SN} \mid M \in \mathcal{T}(\Gamma;\alpha)\}$$

The following result is trivial.

Theorem 8.5. Let $\mathcal{SN}(\Gamma;\alpha) \neq \emptyset$. Then $\mathcal{SN}(\Gamma;\alpha \to \beta) \supset \mathcal{SN}(\Gamma;\alpha) \to \mathcal{SN}(\Gamma;\beta)$.

Proof. Let $M \in \mathcal{SN}(\Gamma;\alpha) \to \mathcal{SN}(\Gamma;\beta)$. Since $\mathcal{SN}(\Gamma,\alpha) \neq \emptyset$, there is some $N \in \mathcal{SN}(\Gamma,\alpha)$. We have

1. $M \in \mathcal{SN}$, because $MN \in \mathcal{SN}(\Gamma;\beta)$,

2. $\Gamma \vdash M : \alpha \to \beta$, by the previous lemma.

□

The converse of this theorem is not so easy and we need the following lemma to prove it.

Lemma 8.6. Let $N \in \mathcal{SN}(\Gamma;\alpha_1) \to \ldots \to \mathcal{SN}(\Gamma;\alpha_n)$ with α_n a base type. Let $P \in \mathcal{SN}(\Gamma';\beta)$ with $\Gamma' = \Gamma, x : \alpha_1 \to \ldots \to \alpha_n$. Then $P[x := N] \in \mathcal{SN}(\Gamma;\beta)$.

Proof. We make use of the Substitution Lemma for $\lambda^{\to}$ à la Curry. It states that $\Gamma, x : \beta \vdash M : \alpha$ and $\Gamma \vdash N : \beta$ implies $\Gamma \vdash M[x := N] : \alpha$.

The proof proceeds by induction on the derivation of $P \in \mathcal{SN}$.

1. Suppose $P = yP_1 \ldots P_k$ with $P_1, \ldots, P_k \in \mathcal{SN}$. By induction hypothesis, we have $P_i[x := N] \in \mathcal{SN}$ for $i = 1, \ldots, k$. We write P_i^* for $P_i[x := N]$ for $i = 1, \ldots, k$.

   If $y \neq x$, then $P[x := N] \in \mathcal{SN}$ follows from the fact that $P_i^* \in \mathcal{SN}$ for $i = 1, \ldots, k$. Using the Substitution Lemma we obtain $P[x := N] \in \mathcal{SN}(\Gamma;\beta)$.

   If $y = x$, then we have to prove that $NP_1^* \ldots P_k^* \in \mathcal{SN}(\Gamma;\beta)$. We have $k \leq n$ and by the induction hypothesis and the Substitution Lemma $P_i^* \in \mathcal{SN}(\Gamma;\alpha_i)$ for $i = 1, \ldots, k$. Further, $N \in \mathcal{SN}(\Gamma;\alpha_1) \to \ldots \to \mathcal{SN}(\Gamma;\alpha_n) \subset \mathcal{SN}(\Gamma;\alpha_1) \to \ldots \to \mathcal{SN}(\Gamma;\alpha_k) \to \mathcal{SN}(\Gamma;\beta)$, by Theorem 8.5. Hence we have $P[x := N] = NP_1^* \ldots P_k^* \in \mathcal{SN}(\Gamma;\beta)$.

2. Suppose $P = \lambda y.P_0$ with $P_0 \in \mathcal{SN}$. By induction hypothesis, we have $P_0[x := N] \in \mathcal{SN}$. Together with the Substitution Lemma this yields $P[x := N] = (\lambda z.P_0)[x := N] \in \mathcal{SN}(\Gamma,\beta)$.

3. Suppose $P = (\lambda y.P_0)P_1P_2 \ldots P_k$ with $P_0[y := P_1]P_2 \ldots P_k \in \mathcal{SN}$ and $P_1 \in \mathcal{SN}$. By induction hypothesis we have $(P_0[y := P_1]P_2 \ldots P_k)[x := N] \in \mathcal{SN}$ and $P_1[x := N] \in \mathcal{SN}$. Using the Substitution Lemma this yields $P[x := N] = ((\lambda y.P_0)P_1P_2 \ldots P_k)[x := N] =\in \mathcal{SN}(\Gamma;\beta)$.

□

Now we can prove the following theorem.

THEOREM 8.7. $\mathcal{SN}(\Gamma;\alpha \to \beta) \subset \mathcal{SN}(\Gamma;\alpha) \to \mathcal{SN}(\Gamma;\beta)$.

Proof. Let $M \in \mathcal{SN}(\Gamma;\alpha \to \beta)$. We prove that for all $N \in \mathcal{SN}(\Gamma;\alpha)$, we have $MN \in \mathcal{SN}(\Gamma;\beta)$. Let thereto $N \in \mathcal{SN}(\Gamma;\alpha)$. Note that it is clear that $\Gamma \vdash MN : \beta$. It remains to prove that $MN \in \mathcal{SN}$. This is proven by induction on α and for each α by induction on the derivation of $M \in \mathcal{SN}$.

α **is a base type.** The proof of this part proceeds by induction on the derivation of $M \in \mathcal{SN}$.

1. Suppose $M = xM_1 \ldots M_k$ with $M_1, \ldots, M_k \in \mathcal{SN}$. We have $N \in \mathcal{SN}$ because $N \in \mathcal{SN}(\Gamma;\alpha)$. This yields $MN = xM_1 \ldots M_k N \in \mathcal{SN}$.

2. Suppose $M = \lambda x.P$ with $P \in \mathcal{SN}$. Note that $\Gamma, x : \alpha \vdash P : \beta$, so actually $P \in \mathcal{SN}(\Gamma, x : \alpha; \beta)$. For proving $(\lambda x.P)N \in \mathcal{SN}$, we need to prove $P[x := N] \in \mathcal{SN}$. This follows from an application of Lemma 8.6.

3. Suppose $M = (\lambda x.M_0)M_1M_2 \ldots M_k$ with $M_0[x := M_1]M_2 \ldots M_k \in \mathcal{SN}$ and $M_1 \in \mathcal{SN}$. By induction hypothesis of the induction on the derivation of $M \in \mathcal{SN}$, we have $M_0[x := M_1]M_2 \ldots M_k N \in \mathcal{SN}$. Moreover $M_1 \in \mathcal{SN}$. This yields $(\lambda x.M_0)M_1M_2 \ldots M_k N \in \mathcal{SN}$.

α **is a composed type.** The proof of this part proceeds as well by induction on the derivation of $M \in \mathcal{SN}$.

1. Suppose $M = xM_1 \ldots M_k$ with $M_1, \ldots, M_k \in \mathcal{SN}$. Since $N \in \mathcal{SN}$, we have $MN \in \mathcal{SN}$.

2. Suppose $M = \lambda x.P$ with $P \in \mathcal{SN}$. For proving $(\lambda x.P)N \in \mathcal{SN}$, we need to prove that $P[x := N] \in \mathcal{SN}$. We have $\alpha = \alpha_1 \to \ldots \to \alpha_n$ with α_n a base type. By the induction hypothesis of the induction on α, we have $N \in \mathcal{SN}(\Gamma;\alpha_1) \to \ldots \to \mathcal{SN}(\Gamma;\alpha_n)$. Lemma 8.6 yields that $P[x := N] \in \mathcal{SN}$.

3. Suppose $M = (\lambda x.M_0)M_1M_2 \ldots M_k$ with $M_0[x := M_1]M_2 \ldots M_k \in \mathcal{SN}$ and $M_1 \in \mathcal{SN}$. By induction hypothesis of the induction on the derivation of $M \in \mathcal{SN}$, we have $M_0[x := M_1]M_2 \ldots M_k N \in \mathcal{SN}$. Moreover $M_1 \in \mathcal{SN}$. This yields $MN \in \mathcal{SN}$.

□

COROLLARY 8.8. $\mathcal{SN}(\Gamma;\alpha \to \beta) = \mathcal{SN}(\Gamma;\alpha) \to \mathcal{SN}(\Gamma;\beta)$.

THEOREM 8.9. If $\Gamma \vdash M : \alpha$ then $M \in \mathcal{SN}(\Gamma;\alpha)$.

Proof. The proof proceeds by induction on the derivation of $\Gamma \vdash M : \alpha$.

1. Suppose $\Gamma, x : \alpha \vdash x : \alpha$. A tautology.

2. Suppose $M = \lambda x.P : \beta \to \gamma$ so the last step in the derivation of $\Gamma \vdash M : \alpha$ is an application of the abstraction clause. By induction hypothesis, we have $P \in \mathcal{SN}(\Gamma, x : \beta; \gamma)$. This yields $(\lambda x.P) \in \mathcal{SN}(\Gamma; \alpha \to \beta)$.

3. Suppose $M = PQ$ so the last step in the derivation of $\Gamma \vdash M : \alpha$ follows from the application clause. Suppose $P : \beta \to \gamma$ and $Q : \beta$. By induction hypothesis, $P \in \mathcal{SN}(\Gamma; \beta \to \gamma)$ and $Q \in \mathcal{SN}(\Gamma; \beta)$. By the previous theorem we have $\mathcal{SN}(\Gamma; \beta \to \gamma) = \mathcal{SN}(\Gamma; \beta) \to \mathcal{SN}(\Gamma; \gamma)$. Therefore $PQ \in \mathcal{SN}(\Gamma; \gamma)$.

□

9. λ-calculus with Intersection Types

In this section we compare our characterisation of strongly normalising terms with another characterisation using intersection types. We prove that all terms in our set $\mathcal{SN}$ are typable with intersection types, and vice versa, that all terms that are intersection typable are in our set. First we shortly recall the definition of λ-calculus with intersection types. We consider the system without the type Ω and without the relation $\leq$ on types. Types are built from type variables and from two binary constructors $\to$ and $\wedge$.

The type inference system is given by the following rules:

Start $\quad \Gamma \vdash x : \alpha$ if $x : \alpha \in \Gamma$

$\to$-Introduction $\quad \dfrac{\Gamma, x : \alpha \vdash M : \beta}{\Gamma \vdash \lambda x.M : \alpha \to \beta}$

$\to$-Elimination $\quad \dfrac{\Gamma \vdash M : \alpha \to \beta \quad \Gamma \vdash N : \alpha}{\Gamma \vdash MN : \beta}$

$\wedge$-Introduction $\quad \dfrac{\Gamma \vdash M : \alpha \quad \Gamma \vdash M : \beta}{\Gamma \vdash M : \alpha \wedge \beta}$

$\wedge$-Elimination $\quad \dfrac{\Gamma \vdash M : \alpha \wedge \beta}{\Gamma \vdash M : \alpha \quad \Gamma \vdash M : \beta}$

For the proof of the first result of this section we use the following notation:

$$
\begin{array}{rcl}
\{x : \alpha\} \wedge \Gamma & = & \Gamma_1, x : \alpha \wedge \beta, \Gamma_2 \quad \text{if } \Gamma = \Gamma_1, x : \beta, \Gamma_2 \\
\{x : \alpha\} \wedge \Gamma & = & x : \alpha, \Gamma \quad \text{otherwise}
\end{array}
$$

Then $\Gamma \wedge \Gamma'$ is defined by induction on Γ.

We will make use of the following proposition that is proved in Chapter IV of [Kri93].

Proposition 9.1. Suppose $\Gamma \vdash M[x := N] : \alpha$ and $\Gamma \vdash N : \beta$. Suppose x doesn't occur in Γ. Then $\Gamma, x : \beta \vdash M : \alpha$.

Theorem 9.2. If $M \in \mathcal{SN}$ then there exist a sequent Γ and a type α such that $\Gamma \vdash M : \alpha$.

Proof. The proof proceeds by induction on the derivation of $M \in \mathcal{SN}$.

- Let $M = xP_1 \ldots P_n$ with $P_i \in \mathcal{SN}$ for $i = 1, \ldots, n$. By induction hypothesis there exist for $i = 1, \ldots, n$ a sequent Γ_i and a type α_i such that $\Gamma_i \vdash P_i : \alpha_i$. Define $\Gamma := (x : \alpha_1 \to \ldots \to \alpha_n \to \beta) \wedge \Gamma_1 \wedge \ldots \wedge \Gamma_n$. Then $\Gamma \vdash M : \beta$.

- Let $M = \lambda x.P$ with $P \in \mathcal{SN}$. By induction hypothesis there exist a sequent Γ and a type β with $\Gamma \vdash P : \beta$. Then $\Gamma' \vdash \lambda x.P : \alpha \to \beta$, with Γ' obtained from Γ by removing a possible type declaration for x.

- Let $M = (\lambda x.P)QP_1 \dots P_n$ with $P[x := Q]P_1 \dots P_n \in \mathcal{SN}$ and $Q \in \mathcal{SN}$. By induction hypothesis, there exist a sequent Γ_1 and a type α such that $\Gamma_1 \vdash P_0[x := P_1]P_2 \dots P_n : \alpha$ and there exist a sequent Γ_2 and a type β such that $\Gamma_2 \vdash P_1 : \beta$. Let $\Gamma := \Gamma_1 \wedge \Gamma_2$. Then we have $\Gamma \vdash P_0[x := P_1]P_2 \dots P_n : \alpha$ and $\Gamma \vdash P_1 : \beta$. Moreover, $\Gamma \vdash P_0[x := P_1] : \gamma$ for the appropriate type γ. There are two possibilities: $x \in \mathsf{FV}(P_0)$ or $x \notin \mathsf{FV}(P_0)$. In both cases we have $\Gamma, x : \beta \vdash P_0 : \gamma$: in the first case by the previous proposition and in the second case immediately. So $\Gamma \vdash \lambda x.P_0 : \beta \to \gamma$ and hence $\Gamma \vdash (\lambda x.P_0)P_1 : \gamma$. We can conclude $\Gamma \vdash M : \alpha$.

□

For proving the converse statement we make use of the characterisation of strongly normalising terms. Like in the previous section, we consider the intersection of the set of terms that are typable with a certain type in a certain context and the set $\mathcal{SN}$.

Definition 9.3. $\mathcal{T}(\Gamma; \alpha) = \{M \in \Lambda \mid \Gamma \vdash M : \alpha\}$.

Proposition 9.4.

1. Let $\mathcal{T}(\Gamma; \alpha) \neq \emptyset$. Then $\mathcal{T}(\Gamma; \alpha \to \beta) \subset \mathcal{T}(\Gamma; \alpha) \to \mathcal{T}(\Gamma; \beta)$.
2. $\mathcal{T}(\Gamma; \alpha \wedge \beta) = \mathcal{T}(\Gamma; \alpha) \cap \mathcal{T}(\Gamma; \beta)$.

Proof.

1. Let $M \in \mathcal{T}(\Gamma; \alpha \to \beta)$ and $N \in \mathcal{T}(\Gamma; \alpha)$. It is clear that $\Gamma \vdash MN : \beta$.
2. Trivial.

□

The statement $\mathcal{T}(\Gamma; \alpha \to \beta) \supset \mathcal{T}(\Gamma; \alpha) \to \mathcal{T}(\Gamma; \beta)$ is not true. However, in the system with the subtype relation denoted by $\leq$, we can prove the following weaker result provided $\mathcal{T}(\Gamma, \alpha) \neq \emptyset$:

$$\mathcal{T}(\Gamma; \alpha \to \beta') \supset \mathcal{T}(\Gamma; \alpha) \to \mathcal{T}(\Gamma; \beta)$$

where β' is some type with $\beta \leq \beta'$. A counterexample to the stronger statement is as follows. Let $\Gamma = x : (\alpha \to \alpha) \wedge ((\alpha \to \alpha) \to (\alpha \to \alpha))$. Then it is easy to see that $x \in \mathcal{T}(\Gamma; \alpha \wedge (\alpha \to \alpha)) \to \mathcal{T}(\Gamma; \alpha \wedge (\alpha \to \alpha))$. However, we don't have $x \in \mathcal{T}(\Gamma; (\alpha \wedge (\alpha \to \alpha)) \to (\alpha \wedge (\alpha \to \alpha)))$. Note that for the η-expansion $\lambda z.xz$ of x, with $z : \alpha \wedge (\alpha \to \alpha)$ it does hold that $\lambda z.xz \in \mathcal{T}(\Gamma; (\alpha \wedge (\alpha \to \alpha)) \to (\alpha \wedge (\alpha \to \alpha)))$.

Exactly like in the previous section, we define the set $\mathcal{SN}(\Gamma; \alpha)$.

Definition 9.5. $\mathcal{SN}(\Gamma, \alpha) = \{M \in \mathcal{SN} \mid M \in \mathcal{T}(\Gamma; \alpha)\}$.

Our goal is now to prove the following:

1. $\mathcal{SN}(\Gamma, \alpha \wedge \beta) = \mathcal{SN}(\Gamma, \alpha) \cap \mathcal{SN}(\Gamma, \beta)$
2. $\mathcal{SN}(\Gamma, \alpha \to \beta) \subset \mathcal{SN}(\Gamma, \alpha) \to \mathcal{SN}(\Gamma, \beta)$

Then it follows by a straightforward induction that $\Gamma \vdash M : \alpha$ implies that $M \in \mathcal{SN}(\Gamma, \alpha)$ and hence M is strongly normalising. The first point is trivial.

THEOREM 9.6. $\mathcal{SN}(\Gamma, \alpha \wedge \beta) = \mathcal{SN}(\Gamma, \alpha) \cap \mathcal{SN}(\Gamma, \beta)$.

The second point requires more care. We make use of a lemma which is a restricted form of the one used in the case of simply typed λ-calculus. Further, we need a Generation Lemma, which describes the types of the components of an application. For making precise what we mean by 'restricted' we need the following definition.

DEFINITION 9.7. The order $\mathsf{ord}(\alpha)$ of a type α is defined inductively as follows.

1. $\mathsf{ord}(\alpha) = 0$ if α is a type variable,
2. $\mathsf{ord}(\beta \to \gamma) = \mathsf{ord}(\beta) + \mathsf{ord}(\gamma) + 1$,
3. $\mathsf{ord}(\beta \wedge \gamma) = \mathsf{max}\{\mathsf{ord}(\beta), \mathsf{ord}(\gamma)\}$.

LEMMA 9.8. Let α be a type with $\mathsf{ord}(\alpha) = 0$. Let $P \in \mathcal{SN}(\Gamma, x : \alpha; \beta)$. Let $N \in \mathcal{SN}(\Gamma; \alpha)$. Then $P[x := N] \in \mathcal{SN}(\Gamma; \beta)$.

Proof. The proof proceeds by induction on the derivation of $P \in \mathcal{SN}$. All cases are trivial, because if $P = yP_1 \ldots P_n$ with $n > 0$, it cannot be the case that $x = y$. $\Box$

LEMMA 9.9. Let $\Gamma \vdash MN : \beta$. Then there exist $\alpha_1, \ldots, \alpha_n$ and $\beta_1, \ldots, \beta_n$ such that

$$\begin{array}{rcl} \Gamma & \vdash & M : (\alpha_1 \to \beta_1) \wedge \ldots \wedge (\alpha_n \to \beta_n) \\ \Gamma & \vdash & N : \alpha_1 \wedge \ldots \wedge \alpha_n \\ \beta & = & \beta_1 \wedge \ldots \wedge \beta_n \end{array}$$

Proof. If there is a derivation of $\Gamma \vdash MN : \beta$, then there is a subderivation of at least length one where each conclusion is of the form $\Gamma' \vdash MN : \beta'$. The proof proceeds by induction on the length of this subderivation.

1. Suppose the subderivation is of length one. Then $\Gamma \vdash MN : \beta$ is due to the $\to$-Elimination rule. Then indeed $\Gamma \vdash M : \alpha \to \beta$ and $\Gamma \vdash N : \alpha$ for some type α.
2. If the subderivation is of length greater than one, then $\Gamma \vdash MN : \beta$ can be due to either the $\wedge$-Introduction rule or the $\wedge$-Elimination rule.

   In the first case, we have $\beta = \beta' \wedge \beta''$ and

   $$\frac{\Gamma \vdash MN : \beta' \quad \Gamma \vdash MN : \beta''}{\Gamma \vdash MN : \beta' \wedge \beta''}$$

is the last step in the subderivation we consider. It is easily seen that the statement follows from the induction hypothesis.

In the second case, we have

$$\frac{\Gamma \vdash MN : \beta \wedge \beta'}{\Gamma \vdash MN : \beta}$$

as the last step in the subderivation we consider. Again, a simple application of the induction hypothesis yields the desired result.

□

LEMMA 9.10. Let $\Gamma \vdash MN_1 \ldots N_m : \beta$. For all i such that $1 \leq i \leq m$, there are $\alpha^i_1, \ldots \alpha^i_{m_i}$ and for all i such that $0 \leq i \leq m$, there are $\gamma^i_1, \ldots \gamma^i_{m_i}$ such that for all i with $1 \leq i \leq m$, we have

$$\begin{array}{rcl} \Gamma & \vdash & MN_1 \ldots N_{i-1} : \gamma_{i-1} \\ \Gamma & \vdash & N_i : \alpha^i_1 \wedge \ldots \wedge \alpha^i_{n_i} \\ \gamma_{i-1} & = & (\alpha^i_1 \to \gamma^i_1) \wedge \ldots \wedge (\alpha^i_{n_i} \to \gamma^i_{n_i}) \\ \gamma_i & = & \gamma^i_1 \wedge \ldots \wedge \gamma^i_{n_i} \end{array}$$

Moreover, we have

$$\Gamma \ \vdash \ MN_1 \ldots N_m : \gamma^m_1 \wedge \ldots \gamma^m_{n_m}$$

Proof. For $i = n$ the statement follows from the previous lemma. Suppose the statement holds for $i = p$, then it holds for $i = p - 1$ by the previous lemma (here $p \geq 1$). □

REMARK 9.11. In the situation of Lemma 9.10, for i such that $1 \leq i \leq m$, we have that $\mathsf{ord}(\gamma_{i-1}) > \mathsf{ord}(\gamma_i)$.

LEMMA 9.12. If $\Gamma \vdash x : \beta$ and $x : \alpha$ is in Γ, then $\mathsf{ord}(\beta) \leq \mathsf{ord}(\alpha)$.

Proof. The lemma is proved by an easy induction on the derivation of $\Gamma \vdash x : \beta$. □

Now we prove the following crucial result.

THEOREM 9.13. Let $\mathcal{SN}(\Gamma; \alpha) \neq \emptyset$. Then $\mathcal{SN}(\Gamma; \alpha \to \beta) \subset \mathcal{SN}(\Gamma; \alpha) \to \mathcal{SN}(\Gamma; \beta)$.

Proof. Let $M \in \mathcal{SN}(\Gamma; \alpha \to \beta)$. Let $N \in \mathcal{SN}(\Gamma; \alpha)$. We prove $MN \in \mathcal{SN}(\Gamma; \beta)$. Note that clearly $MN \in \mathcal{T}(\Gamma; \beta)$ so it remains to show that $MN \in \mathcal{SN}$. The proof proceeds by induction on $\mathsf{ord}(\alpha)$.

$\mathsf{ord}(\alpha) = 0$. The proof of the base step is by induction on the derivation of $M \in \mathcal{SN}$.

1. Let $M = xM_1 \ldots M_n$ with $M_1, \ldots, M_n \in \mathcal{SN}$ Then $MN \in \mathcal{SN}$ by definition of $\mathcal{SN}$.
2. Let $M = \lambda x.P$ with $P \in \mathcal{SN}$. We need to prove $P[x := N]$. This is the case by Lemma 9.8.
3. Let $M = (\lambda x.M_0)M_1M_2 \ldots M_n$ with $M_0[x := M_1]M_2 \ldots M_n \in \mathcal{SN}$ and $M_1 \in \mathcal{SN}$. It follows by induction hypothesis of the induction on the derivation of $M \in \mathcal{SN}$ that $MN \in \mathcal{SN}$.

$\mathsf{ord}(\alpha) > 0$. The proof of the induction step is also given by induction on the derivation of $M \in \mathcal{SN}$.

1. Let $M = xM_1 \dots M_n$ with $M_1, \dots, M_n \in \mathcal{SN}$ Then $MN \in \mathcal{SN}$ by definition of $\mathcal{SN}$.

2. Let $M = \lambda x.P$ with $P \in \mathcal{SN}$. This is the most complicated case. It is proved by induction on the derivation of $M \in \mathcal{SN}$. The problematic case in this induction is if $M = yP_1 \dots P_m$ with $m > 0$ and $x = y$. We only consider the problematic case since the other cases follow easily by induction. Let P_i^* denote $P_i[x := N]$.

   We have $\Gamma \vdash NP_1^* \dots P_m^* : \beta$, and we are in the situation of Lemma 9.10. That means, for i with $1 \leq i \leq m$ there are $\alpha_1^i, \dots, \alpha_{m_i}^i$ and for i with $0 \leq i \leq m$ there are $\gamma_1^1, \dots, \gamma_{m_i}^i$ such that for i with $1 \leq i \leq m$ the following holds:

$$\begin{array}{rcl} \Gamma & \vdash & NP_1^* \dots P_{i-1}^* : \gamma_{i-1} \\ \Gamma & \vdash & P_i^* : \alpha_1^i \wedge \dots \wedge \alpha_{p_i}^i \\ \gamma_{i-1} & = & (\alpha_1^i \to \gamma_1^i) \wedge \dots \wedge (\alpha_{p_i}^i \to \gamma_{p_i}^i) \\ \gamma_i & = & \gamma_1^i \wedge \dots \wedge \gamma_{p_i}^i \end{array}$$

   with, moreover, $\Gamma \vdash NP_1^* \dots P_m^* : \gamma_1^m \wedge \dots \wedge \gamma_{p_m}^m$.

   By Lemma 9.12 we have $\mathsf{ord}(\alpha) \geq \mathsf{ord}(\gamma_0)$. Furthermore, we have that $\mathsf{ord}(\gamma_{i-1}) > \mathsf{ord}(\gamma_i)$ for i with $1 \leq i \leq m$.

   This yields that for all i with $1 \leq i \leq m$ and for all j with $1 \leq j \leq p_i$ we have that $\mathsf{ord}(\alpha) > \mathsf{ord}(\alpha_j^i)$. By induction hypothesis we have that $\mathcal{SN}(\Gamma, \alpha_j^i \to \gamma_j^i) \subset \mathcal{SN}(\Gamma, \alpha_j^i) \to \mathcal{SN}(\Gamma, \gamma_j^i)$. Hence we have that $NP_1^* \dots P_{i-1}^* \in \mathcal{SN}(\Gamma, \alpha_j^i) \to \mathcal{SN}(\Gamma, \gamma_j^i)$.

   Hence we have $P[x := N] \in \mathcal{SN}$ and therefore $MN \in \mathcal{SN}$.

3. Let $M = (\lambda x.M_0)M_1M_2 \dots M_n$ with $M_0[x := M_1]M_2 \dots M_n \in \mathcal{SN}$ and $M_1 \in \mathcal{SN}$. It follows by induction hypothesis of the induction on the derivation of $M \in \mathcal{SN}$ that $MN \in \mathcal{SN}$.

□

The second main result of this section is the following theorem.

THEOREM 9.14. If $\Gamma \vdash M : \alpha$, then $M \in \mathcal{SN}(\Gamma, \alpha)$.

Proof. The proof proceeds by induction on the derivation of $\Gamma \vdash M : \alpha$.

1. Suppose $M = x$ and the derivation of $\Gamma \vdash M : \alpha$ consists just of the start rule. Then the statement trivially holds.

2. Suppose we have $\Gamma \vdash \lambda x.P : \alpha \to \beta$ as a consequence of the $\to$-Introduction rule with hypothesis $\Gamma, x : \alpha \vdash P : \beta$. By induction hypothesis, we have $P \in \mathcal{SN}(\Gamma, x : \alpha; \beta)$. It follows that $\lambda x.P \in \mathcal{SN}(\Gamma, \alpha \to \beta)$.

3. Suppose we have $\Gamma \vdash PQ : \beta$ as a consequence of the $\to$-Elimination rule with hypotheses $\Gamma \vdash P : \alpha \to \beta$ and $\Gamma \vdash Q : \alpha$. By induction hypothesis, we have $P \in \mathcal{SN}(\Gamma; \alpha \to \beta)$ and $Q \in \mathcal{SN}(\Gamma; \alpha)$. Using the previous theorem we obtain $PQ \in \mathcal{SN}(\Gamma; \beta)$.

4. Suppose we have $\Gamma \vdash P : \alpha \wedge \beta$ as a consequence of the $\wedge$-Introduction rule with hypotheses $\Gamma \vdash P : \alpha$ and $\Gamma \vdash P : \beta$. By induction hypothesis, we have $P \in \mathcal{SN}(\Gamma;\alpha)$ and $P \in \mathcal{SN}(\Gamma;\beta)$. Using Theorem 9.6 we obtain $P \in \mathcal{SN}(\Gamma;\alpha) \cap \mathcal{SN}(\Gamma;\beta) = \mathcal{SN}(\Gamma;\alpha \wedge \beta)$.

5. Suppose we have $\Gamma \vdash P : \alpha$ as a consequence of the $\wedge$-Elimination rule with hypothesis $\Gamma \vdash P : \alpha \wedge \beta$. By induction hypothesis, we have $P \in \mathcal{SN}(\Gamma;\alpha \wedge \beta)$. By Theorem 9.6, we have $P \in \mathcal{SN}(\Gamma;\alpha) \cap \mathcal{SN}(\Gamma;\beta)$. So $P \in \mathcal{SN}(\Gamma;\alpha)$.

10. Related Work and Conclusions

In this section we discuss the relation between our proofs and other proofs of the same results.

The set $\mathcal{SN}$. Ralph Loader defines the set $\mathcal{SN}$ in a note distributed on the types mailinglist [Loa95], where he announces a proof of strong normalisation of system F. The definitions must have been given more or less simultaneously.

Developments. The result that all β-developments are finite is a classical result in λ-calculus and various proofs already exist. There is for instance a proof that can be found in [Bar84], that makes use of a decreasing labelling. This proof is not related to ours. There is a short and elegant proof by de Vrijer [dV85], in which an exact bound for the length of a development is computed. For proving that the bound is an exact bound, he makes in fact use of the strategy F_∞. Some small observations concerning developments coincide with some small observations we make use of.

Two perpetual strategies. The original proofs of the facts that F_{bk} and F_∞ are perpetual proceed by a case analysis. In both cases it is proved that $F(M)$ admits an infinite rewrite sequence if M does so. In our proof the equivalent statement $F(M) \in \mathcal{SN} \Rightarrow M \in \mathcal{SN}$ is shown. Proving $F(M) \in \mathcal{SN} \Rightarrow M \in \mathcal{SN}$ and using the definition of $\mathcal{SN}$ make our proofs more perspicuous.

A maximal strategy. The fact that F_∞ is a maximal strategy has been proved by Regnier [Reg94] using a relation that permits to permute redexes. In fact, in the paper Regnier shows that some operational criteria do not permit to distinguish between terms that are equivalent up to some permutation of redexes. Much more in the spirit of the present work is a paper by Sørensen ([Sør94]), who gives a proof that is very similar to ours. His work was developed independently and simultaneously. A difference is that in the present paper the number of steps of an F_∞ rewrite sequence is computed explicitly.

Normalisation of simply typed λ-calculus. Many proofs of strong normalisation of simply typed λ-calculus exist. Our proof seems to be mostly related to the proof by Tait and Girard using saturated sets. There are however some important differences.

First, in the proof by Tait and Girard, a type α is interpreted as a set of λ-terms denoted by $[\![\alpha]\!]$. Then, the interpretation of a type $\alpha \to \beta$ is defined to be $[\![\alpha]\!] \to [\![\beta]\!]$. So $[\![\alpha \to \beta]\!] = [\![\alpha]\!] \to [\![\beta]\!]$ by definition, whereas in our proof we need to prove $\mathcal{T}(\Gamma;\alpha \to \beta) = \mathcal{T}(\Gamma;\alpha) \to \mathcal{T}(\Gamma;\beta)$.

On the other hand, in the proof by Tait and Girard it needs to be proved that $[\![\alpha]\!] \to [\![\beta]\!]$ is a subset of the set of strongly normalising terms. In fact, it is proved that the interpretation of a type is a *saturated set*. A saturated set is a subset X of the set of strongly normalising λ-terms that satisfies

1. if x is a variable and $M_1, \ldots, M_n$ are strongly normalising terms then $xM_1 \ldots M_n \in X$,
2. if $M[x := N]P_1 \ldots P_n \in X$ and N is strongly normalising then $(\lambda x.M)NP_1 \ldots P_n \in X$.

In our proof, the set $\mathcal{SN}(\Gamma; \alpha \to \beta)$ is a subset of the set of strongly normalising terms by definition. But the equality $\mathcal{SN}(\Gamma; \alpha \to \beta) = \mathcal{SN}(\Gamma; \alpha) \to \mathcal{SN}(\Gamma; \beta)$ needs to be proved.

λ-calculus with Intersection Types. Krivine [Kri93] proved that the set of strongly normalising terms coincides with the set of terms that are typable in $\lambda\wedge$. For proving that a term that is typable in $\lambda\wedge$ is strongly normalising he makes use of saturated sets. We again make use of sets $\mathcal{SN}(\Gamma, \alpha)$ that contain all terms that are strongly normalising and that are typable in Γ with type α.

Conclusions. We have presented a characterisation of the set of strongly normalising terms that is intuitive and elegant. Using this set we have given simple proofs of properties concerning normalisation in λ-calculus.

Acknowledgements. We gratefully acknowledge remarks by Morten Heine Sørensen. We thank Herman Geuvers, Jan Willem Klop, Vincent van Oostrom and the participants of the TeReSe seminar at the Vrije Universiteit Amsterdam for valuable discussions.

REFERENCES

[Acz77] Peter Aczel. An introduction to inductive definitions. In Jon Barwise, editor, *Handbook of Mathematical Logic*, pages 739–782. North-Holland, 1977.

[Bar84] H.P. Barendregt. *The Lambda Calculus, its Syntax and Semantics.* North Holland, second edition, 1984.

[BBKV76] Henk Barendregt, Jan Bergstra, Jan Willem Klop, and Henri Volken. Degrees, reductions and representability in the lambda calculus. Technical report, University Utrecht, 1976. Preprint nr. 22.

[BK82] Jan A. Bergstra and Jan Willem Klop. Strong normalization and perpetual reductions in the lambda calculus. *Journal of Information Processing and Cybernetics*, 18:403–417, 1982.

[dV85] Roel de Vrijer. A direct proof of the finite developments theorem. *Journal of Symbolic Logic*, 50:339–343, 1985.

[Gir72] J.-Y. Girard. *Interprétation fonctionelle et élimination des coupures de l'arithmétique d'ordre supérieur.* PhD thesis, Université Paris VII, 1972.

[Kri93] J.L. Krivine. *Lambda-Calculus, Types and Models.* Masson, 1993.

[Lév78] Jean-Jacques Lévy. *Réductions correctes et optimales dans le lambda-calcul.* PhD thesis, Université de Paris VII, 1978.

[Loa95] R. Loader. Normalisation by translation. Note distributed on the 'types' mailinglist, April 1995.

[Oos94] Vincent van Oostrom. *Confluence for Abstract and Higher-Order Rewriting.* PhD thesis, Vrije Universiteit, Amsterdam, March 1994.

[Raa93] F. van Raamsdonk. Confluence and superdevelopments. In C. Kirchner, editor, *Proceedings of the 5th International Conference on Rewrite Techniques and Applications*, 1993.

[Reg94] Laurent Regnier. Une équivalence sur les lambda-termes. *Theoretical Computer Science*, 126:281–292, 1994.

[Sør94] Morten Heine Sørensen. Effective longest reduction paths in untyped λ-calculus, 1994. Draft, Katholieke Universiteit Nijmegen.

[Tai67] W.W. Tait. Intensional interpretation of functionals of finite type I. *Journal of Symbolic Logic*, 32:198–212, 1967.

[Ter94] Jan Terlouw. Inductieve definities. Technical report, Rijksuniversiteit Groningen, 1994.